Finding independent sets in hereditary graph classes

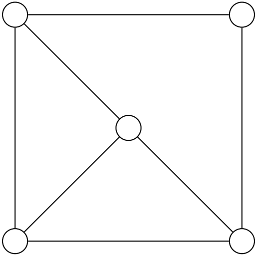
Rémi Watrigant
(joint work with É. Bonnet, N. Bousquet, P. Charbit, S. Thomassé)

Université de Lyon - École Normale Supérieure de Lyon, France

CALDAM 2020 pre-conference school
February 10-11, 2020. Hyderabad, India

Maximum Independent Set (MIS)

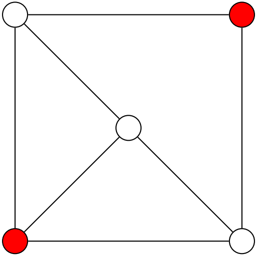
Problem: Given a graph



and an integer k: Is there an independent set of size at least k?

Maximum Independent Set (MIS)

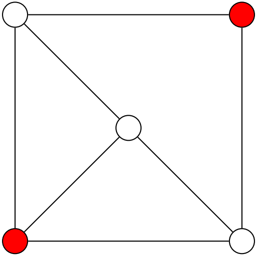
Problem: Given a graph



and an integer k: Is there an independent set of size at least k?

Maximum Independent Set (MIS)

Problem: Given a graph



and an integer k: Is there an independent set of size at least k?

- classical **NP-complete** problem
- $W[1]$-complete (no algorithm running in $f(k)n^{O(1)}$ time)
- no $n^{1-\varepsilon}$-approximation unless $P = NP$

What about on restricted graphs classes?

Maximum Independent Set (MIS) in restricted graph classes

Among others:

- polynomial in bipartite graphs (Kőnig's theorem)
- polynomial in chordal graphs (simplicial decomposition)
- polynomial in perfect graphs (ellipsoid method)
- polynomial in P_6-free graphs [Grzesik et al, SODA 19]
- NP-complete in graphs with maximum degree 3
- NP-complete in triangle-free graphs

Maximum Independent Set (MIS) in restricted graph classes

Among others:

- polynomial in bipartite graphs (Kőnig's theorem)
- polynomial in chordal graphs (simplicial decomposition)
- polynomial in perfect graphs (ellipsoid method)
- polynomial in P_6-free graphs [Grzesik et al, SODA 19]
- NP-complete in graphs with maximum degree 3
- NP-complete in triangle-free graphs

General question:

Given a fixed graph H, can we solve Maximum Independent Set more efficiently in H-free graphs[a]?

[a] Graphs that do not admit H as an **induced subgraph**.

Maximum Independent Set (MIS) in restricted graph classes

Among others:

- polynomial in bipartite graphs (Kőnig's theorem)
- polynomial in chordal graphs (simplicial decomposition)
- polynomial in perfect graphs (ellipsoid method)
- polynomial in P_6-free graphs [Grzesik et al, SODA 19]
- NP-complete in graphs with maximum degree 3
- NP-complete in triangle-free graphs

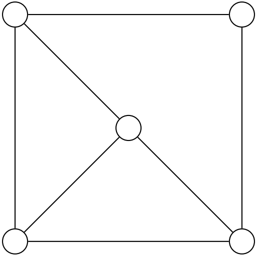
General question:

Given a fixed graph H, can we solve Maximum Independent Set more efficiently in H-free graphs[a]?

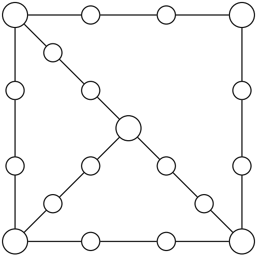
[a] Graphs that do not admit H as an **induced subgraph**.

"more efficiently": polynomial? approximation? FPT?

NP-hard cases [Poljak, 1974, Alekseev '82]

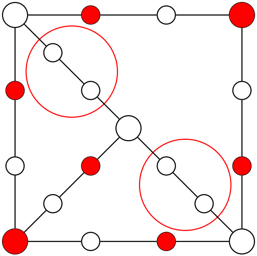


NP-hard cases [Poljak, 1974, Alekseev '82]



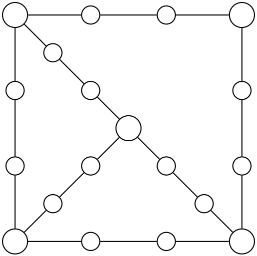
Subdivide every edge twice

NP-hard cases [Poljak, 1974, Alekseev '82]



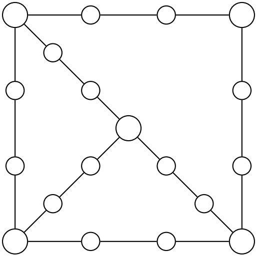
Subdivide every edge twice $\Rightarrow \alpha(G') = \alpha(G) + |E(G)|$

NP-hard cases [Poljak, 1974, Alekseev '82]



Subdivide every edge any fixed even number of times

NP-hard cases [Poljak, 1974, Alekseev '82]

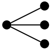


Subdivide every edge any fixed even number of times

Maximum Independent Set remains NP-hard in H-free graphs except if H is...

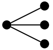
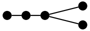
P/NP-complete status of MIS on H-free graphs

Maximum Independent Set remains NP-hard in H-free graphs except if H is...
for H <u>connected</u>:

- NP-complete, if H is **<u>not</u>** a path or a subdivided claw (claw = )
  → **"most" graphs**

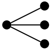
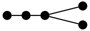
P/NP-complete status of MIS on H-free graphs

Maximum Independent Set remains NP-hard in H-free graphs except if H is...
for H <u>connected</u>:

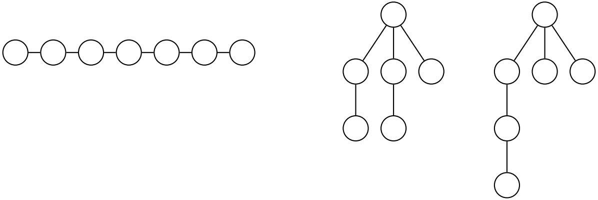
- NP-complete, if H is **<u>not</u>** a path or a subdivided claw (claw = )
  $\rightarrow$ **“most” graphs**
- in P, if H is a path on up to 6 vertices
- in P, if H is a claw with one edge subdivided once ( a.k.a. the fork)

P/NP-complete status of MIS on H-free graphs

Maximum Independent Set remains NP-hard in H-free graphs except if H is...
for H connected:

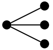
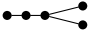
- NP-complete, if H is **not** a path or a subdivided claw (claw = )
  → **"most" graphs**
- in P, if H is a path on up to 6 vertices
- in P, if H is a claw with one edge subdivided once ( a.k.a. the fork)
- for other H, the problem is open

Minimal open cases:



P/NP-complete status of MIS on H-free graphs

Maximum Independent Set remains NP-hard in H-free graphs except if H is...
for H <u>connected</u>:

- NP-complete, if H is **<u>not</u>** a path or a subdivided claw (claw = )
  $\rightarrow$ **"most" graphs**
- in P, if H is a path on up to 6 vertices
- in P, if H is a claw with one edge subdivided once ( a.k.a. the fork)
- for other H, the problem is open

Here, we focus on NP-complete cases

When a problem is NP-complete:

Parameterized algorithms

A problem is **Fixed-Parameter Tractable** (FPT) if it can be solved in time $f(k)n^{O(1)}$, where

- n is the size of an instance
- k is a parameter (here: size of the independent set we are looking for)
- $f(.)$ is any computable (possibly exponential) function

When a problem is NP-complete:

Parameterized algorithms

A problem is **Fixed-Parameter Tractable** (FPT) if it can be solved in time $f(k)n^{O(1)}$, where

- n is the size of an instance
- k is a parameter (here: size of the independent set we are looking for)
- $f(.)$ is any computable (possibly exponential) function

Do all problems admit FPT algorithms?

- Maximum Independent set can be solved in $O(n^k.k^2)$ time
  $\rightarrow$ **but this is not FPT**

When a problem is NP-complete:

Parameterized algorithms

A problem is **Fixed-Parameter Tractable** (FPT) if it can be solved in time $f(k)n^{O(1)}$, where

- n is the size of an instance
- k is a parameter (here: size of the independent set we are looking for)
- $f(.)$ is any computable (possibly exponential) function

Do all problems admit FPT algorithms?

- Maximum Independent set can be solved in $O(n^k.k^2)$ time
  $\rightarrow$ **but this is not FPT**

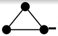
Unless FPT $\neq$ W[1], Maximum Independent Set (in general graphs) is **not** FPT

When a problem is NP-complete:

Parameterized algorithms

A problem is **Fixed-Parameter Tractable** (FPT) if it can be solved in time $f(k)n^{O(1)}$, where

- n is the size of an instance
- k is a parameter (here: size of the independent set we are looking for)
- $f(.)$ is any computable (possibly exponential) function

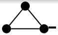
Toy example: Independent Set is FPT in -free graphs:

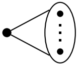
When a problem is NP-complete:

Parameterized algorithms

A problem is **Fixed-Parameter Tractable** (FPT) if it can be solved in time $f(k)n^{O(1)}$, where

- n is the size of an instance
- k is a parameter (here: size of the independent set we are looking for)
- $f(.)$ is any computable (possibly exponential) function

Toy example: Independent Set is FPT in -free graphs:

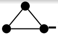
- case 1: there is a vertex of degree $\geq k$: 

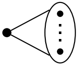
When a problem is NP-complete:

Parameterized algorithms

A problem is **Fixed-Parameter Tractable** (FPT) if it can be solved in time $f(k)n^{O(1)}$, where

- n is the size of an instance
- k is a parameter (here: size of the independent set we are looking for)
- $f(.)$ is any computable (possibly exponential) function

Toy example: Independent Set is FPT in -free graphs:

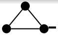
- case 1: there is a vertex of degree $\geq k$:  

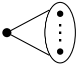
When a problem is NP-complete:

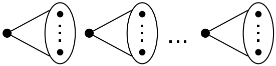
Parameterized algorithms

A problem is **Fixed-Parameter Tractable** (FPT) if it can be solved in time $f(k)n^{O(1)}$, where

- n is the size of an instance
- k is a parameter (here: size of the independent set we are looking for)
- $f(.)$ is any computable (possibly exponential) function

Toy example: Independent Set is FPT in -free graphs:

- case 1: there is a vertex of degree $\geq k$:  
- case 2 : $d(v) < k$ for all $v \in V$:



  - either we construct an independent set of size k
  - or the graph has at most $k(k-1) = O(k^2)$ vertices $\Rightarrow$ brute-force

More generally:

Ramsey's theorem

Given $r, k > 1$, there exists an integer $Ram(r, k)$ such that any graph G with at least $Ram(r, k)$ vertices must contain either:

- a clique of size r, or
- an independent set of size k

Example: $Ram(3, 3) = 6$, $Ram(4, 4) = 18$

More generally:

Ramsey's theorem

Given $r, k > 1$, there exists an integer $Ram(r,k)$ such that any graph G with at least $Ram(r,k)$ vertices must contain either:

- a clique of size r, or
- an independent set of size k

Example: $Ram(3,3) = 6$, $Ram(4,4) = 18$

Application : MIS in K_r-free graphs (no clique of size $\geq k$):

- if G has more than $Ram(r,k)$ vertices, answer YES
- Otherwise: $|V(G)| < Ram(r,k) = O(k^{r-1})$

More generally:

Ramsey's theorem

Given $r, k > 1$, there exists an integer $Ram(r, k)$ such that any graph G with at least $Ram(r, k)$ vertices must contain either:

- a clique of size r, or
- an independent set of size k

Example: $Ram(3, 3) = 6$, $Ram(4, 4) = 18$

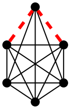
Application : MIS in K_r-free graphs (no clique of size $\geq k$):

- if G has more than $Ram(r, k)$ vertices, answer YES
- Otherwise: $|V(G)| < Ram(r, k) = O(k^{r-1})$

Maximum Independent Set is FPT in K_r-free graphs for every $r \geq 1$

(we say it admits a **kernel** with $O(k^{r-1})$ vertices)

Now : what if the forbidden graph H is "almost" a clique?



Let's call it K_r^{-2}

$\rightarrow$ can't use Ramsey here...

Erdős-Hajnal Conjecture

H satisfies the **Erdős-Hajnal** property if there is $0 < \varepsilon_H \leq 1$ such that for any H-free graph G on n vertices, either:

- G contains a clique of size n^{ε_H}, or
- G contains an independent set of size n^{ε_H}

Conjecture: every graph satisfies the EH-property

Erdős-Hajnal Conjecture

H satisfies the **Erdős-Hajnal** property if there is $0 < \varepsilon_H \leq 1$ such that for any H-free graph G on n vertices, either:

- G contains a clique of size n^{ε_H}, or
- G contains an independent set of size n^{ε_H}

Conjecture: every graph satisfies the EH-property

Graphs known to satisfy the EH-property:

- K_r (simple induction)
- any graph on four vertices
- graphs that can be constructed from them by "substitution operation"

Open for many graphs, in particular:

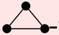
- C_5: cycle on five vertices
- P_5: path on five vertices

Erdős-Hajnal Conjecture

H satisfies the **Erdős-Hajnal** property if there is $0 < \varepsilon_H \leq 1$ such that for any H-free graph G on n vertices, either:

- G contains a clique of size n^{ε_H}, or
- G contains an independent set of size n^{ε_H}

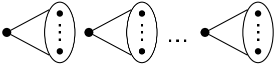
Conjecture: every graph satisfies the EH-property

K_r satisfies the EH-property (e.g. -free graphs have $\alpha(G) \geq \sqrt{n}$)

by induction on r:

- if there is a vertex v of degree $\geq n^{\frac{r-2}{r-1}}$:
  - $N(v)$ is K_{r-1}-free (by induction)

    $\Rightarrow$ there is an independent set of size $\geq \left(n^{\frac{r-2}{r-1}}\right)^{\frac{1}{r-2}} = n^{\frac{1}{r-1}}$
- otherwise:



We construct an independent set of size $\geq \frac{n}{n^{\frac{r-2}{r-1}}} = n^{\frac{1}{r-1}}$

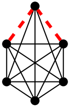
Erdős-Hajnal Conjecture

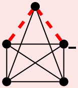
H satisfies the **Erdős-Hajnal** property if there is $0 < \varepsilon_H \leq 1$ such that for any H-free graph G on n vertices, either:

- G contains a clique of size n^{ε_H}, or
- G contains an independent set of size n^{ε_H}

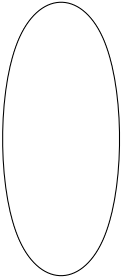
Conjecture: every graph satisfies the EH-property

K_r^{-2} satisfies the Erdős-Hajnal property, and an independent set or a clique of size $n^{\frac{1}{r-1}}$ **can be found in polynomial time** (simple induction)

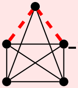


Example: $O(k^4)$ kernel in -free graphs

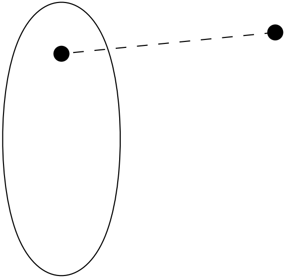
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$

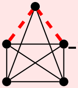


Clique of size $\geq n^{\frac{1}{4}}$

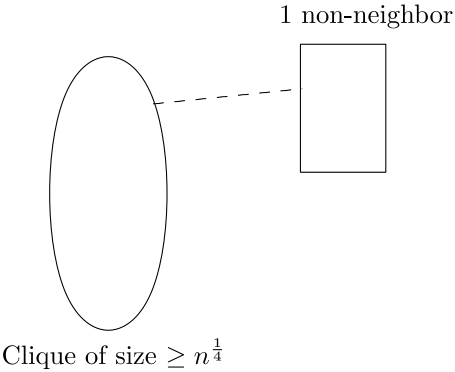
Example: $O(k^4)$ kernel in  -free graphs

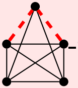
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$



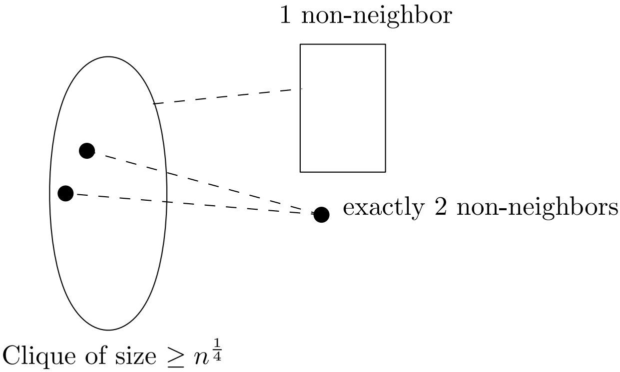
Example: $O(k^4)$ kernel in -free graphs

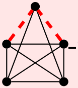
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$



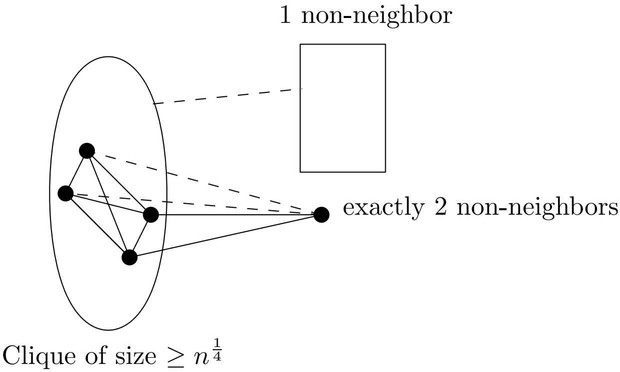
Example: $O(k^4)$ kernel in -free graphs

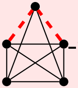
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$



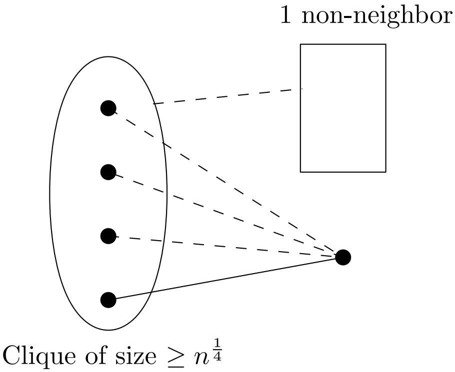
Example: $O(k^4)$ kernel in  -free graphs

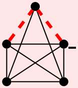
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$



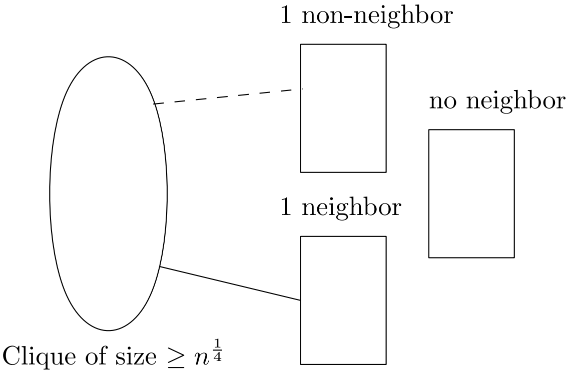
Example: $O(k^4)$ kernel in -free graphs

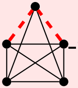
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$



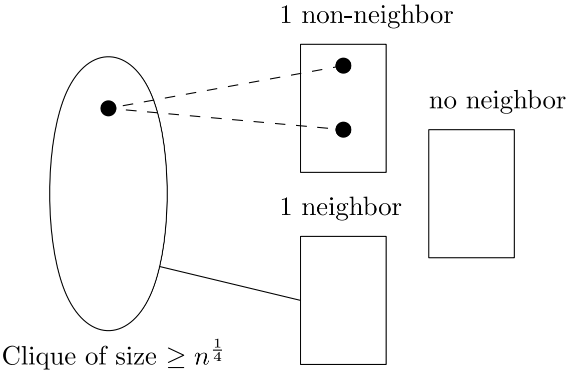
Example: $O(k^4)$ kernel in -free graphs

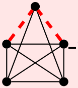
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$



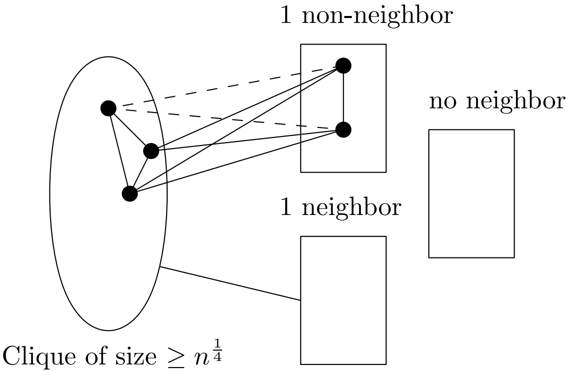
Example: $O(k^4)$ kernel in -free graphs

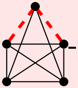
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$



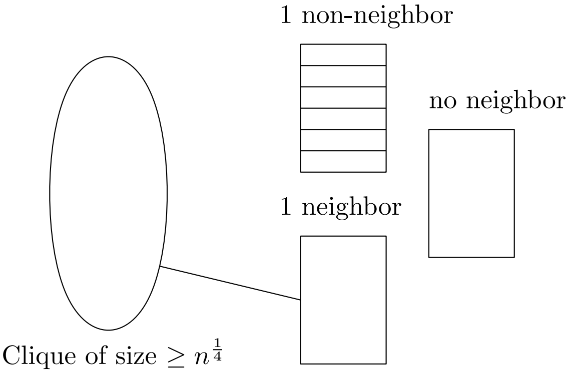
Example: $O(k^4)$ kernel in -free graphs

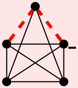
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$



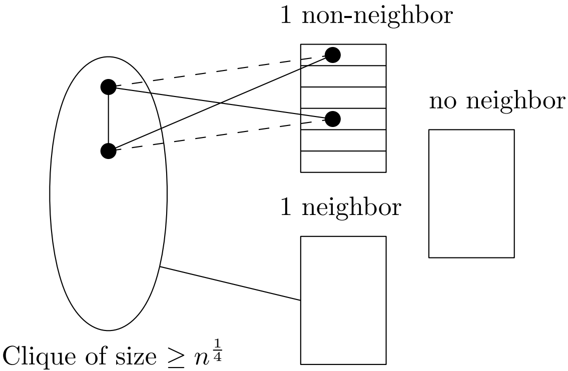
Example: $O(k^4)$ kernel in  -free graphs

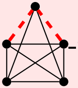
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$



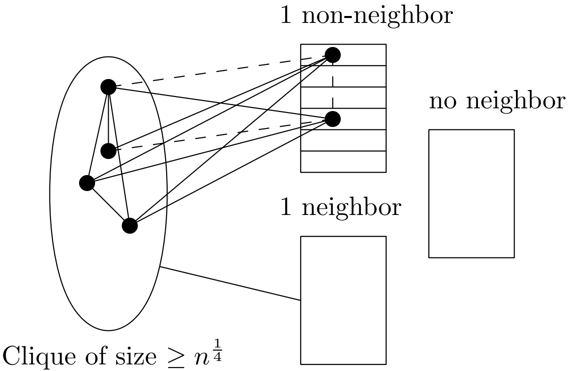
Example: $O(k^4)$ kernel in -free graphs

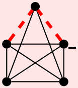
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$



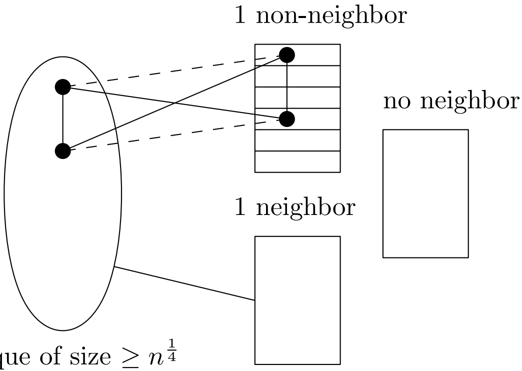
Example: $O(k^4)$ kernel in  -free graphs

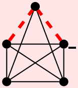
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$



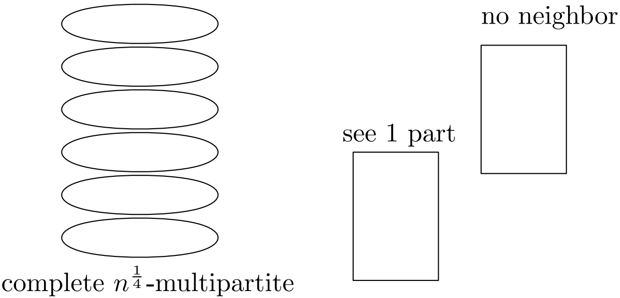
Example: $O(k^4)$ kernel in  -free graphs

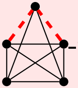
- invoke Erdős-Hajnal algorithm:
  - either large independent set → done
  - or large clique $\geq n^{1/4}$



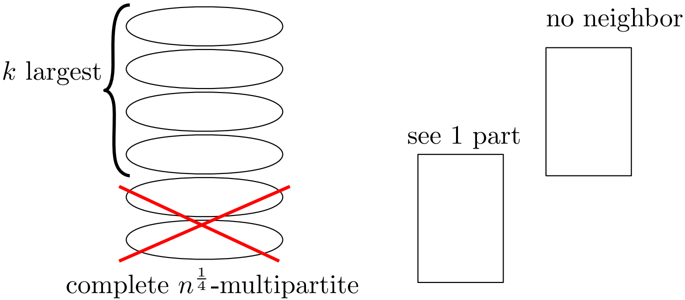
Example: $O(k^4)$ kernel in  -free graphs

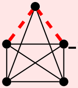
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$



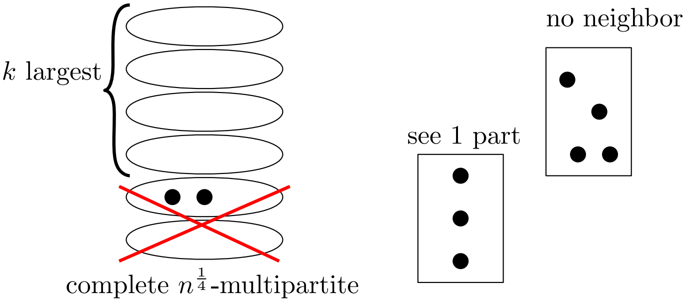
Example: $O(k^4)$ kernel in -free graphs

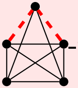
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$
- if $\#parts > k$: **reduce**: keep only the k largest parts



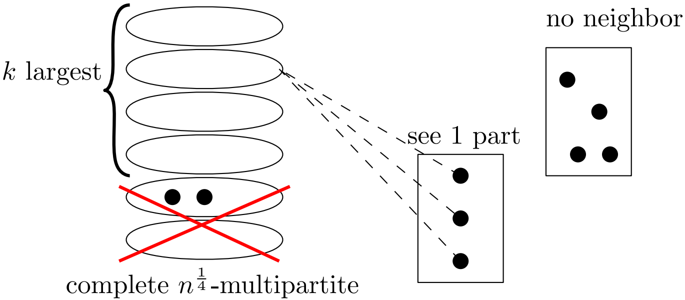
Example: $O(k^4)$ kernel in -free graphs

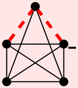
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$
- if $\#\textit{parts} > k$: **reduce**: keep only the k largest parts



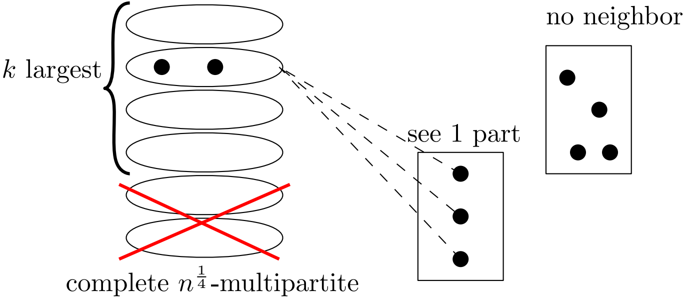
Example: $O(k^4)$ kernel in -free graphs

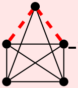
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$
- if $\#parts > k$: **reduce**: keep only the k largest parts



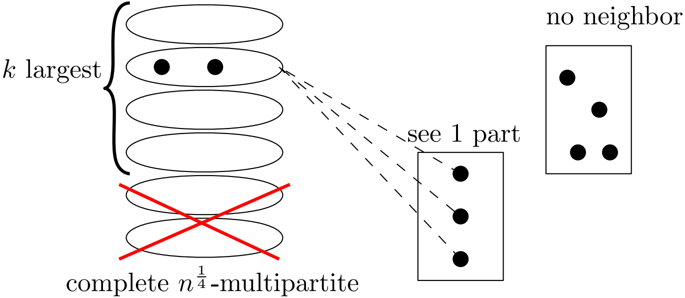
Example: $O(k^4)$ kernel in -free graphs

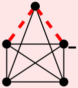
- invoke Erdős-Hajnal algorithm:
  - either large independent set $\to$ done
  - or large clique $\geq n^{1/4}$
- if $\#parts > k$: **reduce**: keep only the k largest parts



Example: $O(k^4)$ kernel in -free graphs

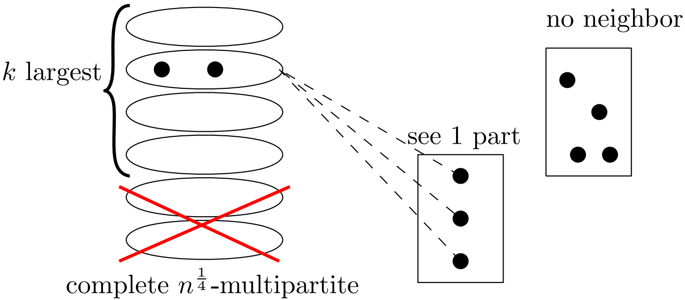
- invoke Erdős-Hajnal algorithm:
  - either large independent set → done
  - or large clique $\geq n^{1/4}$
- if $\#parts > k$: **reduce**: keep only the k largest parts
- at the end:
  $k \geq \#parts$



Example: $O(k^4)$ kernel in -free graphs

- invoke Erdős-Hajnal algorithm:
  - either large independent set $\rightarrow$ done
  - or large clique $\geq n^{1/4}$
- if $\#parts > k$: **reduce**: keep only the k largest parts
- at the end:

  $k \geq \#parts \ \geq n^{1/4} \quad \rightarrow G$ has at most k^4 vertices



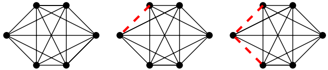
Given a graph H, is computing Maximum Independent Set:

FPT	**not FPT**
$f(k)n^{O(1)}$	W[1]-hard

Given a graph H, is computing Maximum Independent Set:

FPT

$f(k)n^{O(1)}$



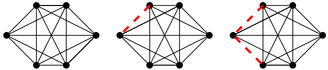
not FPT

W[1]-hard

Given a graph H, is computing Maximum Independent Set:

FPT

$f(k)n^{O(1)}$



not FPT

W[1]-hard

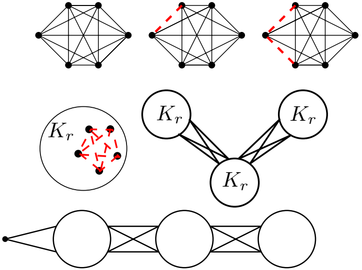
any connected H containing either:

- a cycle of length ≥ 4, or
- $K_{1,4}$, or
- a tree with two vertices of degree ≥ 3

Given a graph H, is computing Maximum Independent Set:

FPT

$f(k)n^{O(1)}$



not FPT

W[1]-hard

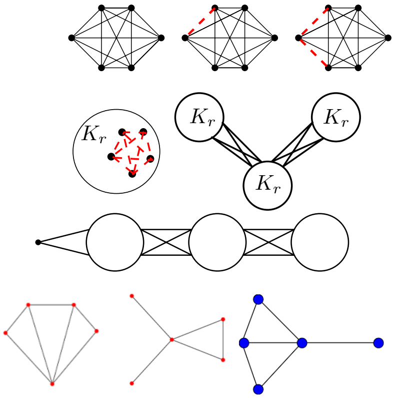
any connected H containing either:

- a cycle of length ≥ 4, or
- $K_{1,4}$, or
- a tree with two vertices of degree ≥ 3

Given a graph H, is computing Maximum Independent Set:

FPT

$f(k)n^{O(1)}$

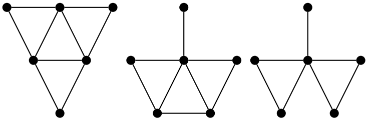


not FPT

W[1]-hard

any connected H containing either:

- a cycle of length ≥ 4, or
- $K_{1,4}$, or
- a tree with two vertices of degree ≥ 3



Open questions

- we thought the FPT/W-hard dichotomy would be an easier job than the P/NP-hard one
  $\rightarrow$ but it looks like a very challenging task

Open questions

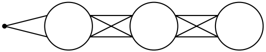
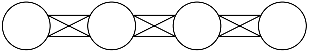
- we thought the FPT/W-hard dichotomy would be an easier job than the P/NP-hard one
  $\rightarrow$ but it looks like a very challenging task

- FPT algorithms for P_t-free graphs?

Open questions

- we thought the FPT/W-hard dichotomy would be an easier job than the P/NP-hard one
  $\rightarrow$ but it looks like a very challenging task

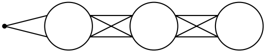
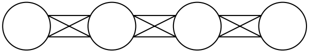
- FPT algorithms for P_t-free graphs?

- generalizing  to 

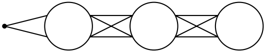
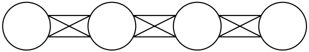
Open questions

- we thought the FPT/W-hard dichotomy would be an easier job than the P/NP-hard one
  $\rightarrow$ but it looks like a very challenging task

- FPT algorithms for P_t-free graphs?

- generalizing  to 
- algorithmic use of structural graph theory results
  - Ramsey
  - Erdős-Hajnal
  - Kővari-Sós-Turán

Open questions

- we thought the FPT/W-hard dichotomy would be an easier job than the P/NP-hard one
  → but it looks like a very challenging task
- FPT algorithms for P_t-free graphs?
- generalizing  to 
- algorithmic use of structural graph theory results
  - Ramsey
  - Erdős-Hajnal
  - Kővari-Sós-Turán
- polynomial kernel/no polynomial kernel dichotomy?

and voilà !

Questions ?