Parameterized Complexity in a Nutshell and Some Meta-Algorithmic Theorems

Christophe PAUL
CNRS - LIRMM

Complex Network Seminar
LIP6 – UPMC
July 8, 2016

Part 1 – An overview of parameterized algorithms

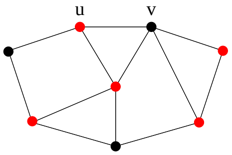
The early steps of NP-Completeness Theory

- Cook's Theorem (1971) : SAT is **NP**-Complete
  (2-SAT $\in$ **P**)
- Karp's list of 21 NP-complete problems (1972), among which:

Common understanding : **NP**–Complete problems are all "equivalent" !

The early steps of NP-Completeness Theory

- Cook's Theorem (1971) : SAT is **NP**-Complete
  (2-SAT $\in$ **P**)
- Karp's list of 21 NP-complete problems (1972), among which:
  - VERTEX COVER: Does there exist a subset S of at most k vertices such that every edge of G is covered by some vertex of S ?



  - INDEPENDENT SET: Does there exist a subset S of at least k vertices pairwise non-adjacent in a graph G ?
  - COLORING: Can the vertices of a graph G be colored by at most k colors in such a way that adjacent vertices receive different colors ?

Common understanding : **NP**–Complete problems are all "equivalent" !

Tractability of **NP**–Complete problems

Observations :

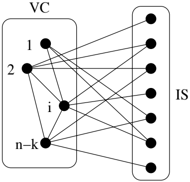
- COLORING is **NP**–Complete for $k = 3$ colors.
- VERTEX COVER and INDEPENDENT SET belong to **P** for every fixed k : naive algorithm in $O(n^k)$ steps.

Tractability of **NP**–Complete problems

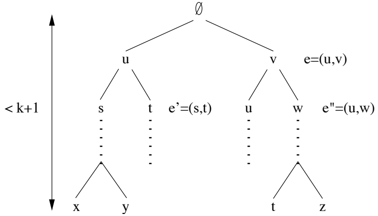
Observations :

- COLORING is **NP**–Complete for $k = 3$ colors.
- VERTEX COVER and INDEPENDENT SET belong to **P** for every fixed k : naive algorithm in $O(n^k)$ steps.

Observation : G has a VERTEX COVER of size k
$\Leftrightarrow$ G has an independent set of size $n - k$



Bounded Search Tree



VERTEX COVER : $O(2^k \cdot (m + n))$

INDEPENDENT SET :
$O(2^{(n-k)} \cdot (m + n))$

Tractability of **NP**–Complete problems

So we have :

- k-Coloring is **Para-NP-Complete**
- k-Independent Set is **XP**
- k-Vertex Cover is **FPT**

Tractability of **NP**–Complete problems

So we have :

- k-Coloring is **Para-NP-Complete**
- k-Independent Set is **XP**
- k-Vertex Cover is **FPT**

Measure the time complexity in term of

- the input size n;
- a parameter k (independent of n) :
  - solution size (natural parameter);
  - maximum degree, treewidth, feedback vertex set size. . . (structural parameters).

Tractability of **NP**–Complete problems

So we have :

- k-COLORING is **Para-NP-Complete**
- k-INDEPENDENT SET is **XP**
- k-VERTEX COVER is **FPT**

Measure the time complexity in term of

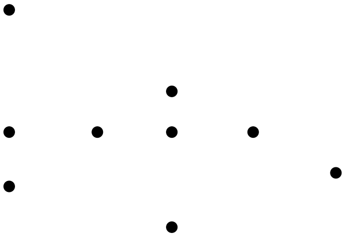
- the input size n;
- a parameter k (independent of n) :
  - solution size (natural parameter);
  - maximum degree, treewidth, feedback vertex set size. . . (structural parameters).

Définition : A parameterized problem is **FPT** (Fixed Parameter Tractable) if it can be solved in $f(k) \cdot n^{O(1)}$ steps.

Observation : the parameter dependence could be : $f(k) = 2^{k^{k^{k^{\cdot^{\cdot^{\cdot^{k^k}}}}}}}$

Polynomial time pre-processing

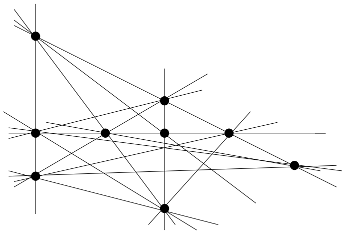
- Polynomial-time pre-processing often aims at reducing the input size. How can we measure the reduction ?



Does there exist k lines covering the set S of points ?

Polynomial time pre-processing

- Polynomial-time pre-processing often aims at reducing the input size. How can we measure the reduction ?

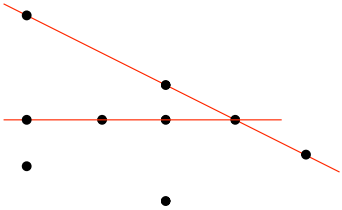


Does there exist k lines covering the set S of points ?

Observation 1: Only lines generated by pair of points of S are relevant.

Polynomial time pre-processing

- Polynomial-time pre-processing often aims at reducing the input size. How can we measure the reduction ?



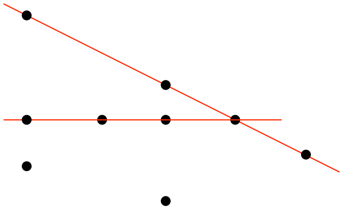
Does there exist k lines covering the set S of points ?

Observation 2: If a line L contains at least $k+1$ points, then it has to belong to the solution (if it exists) (e.g. here $k=3$)

$\Rightarrow$ remove L and decrease k by 1.

Polynomial time pre-processing

- Polynomial-time pre-processing often aims at reducing the input size. How can we measure the reduction ?



Does there exist k lines covering the set S of points ?

Observation 2: If a line L contains at least $k+1$ points, then it has to belong to the solution (if it exists) (e.g. here $k=3$)

$\Rightarrow$ remove L and decrease k by 1.

$\Rightarrow$ a reduced instance contains at most k^2 points.

Kernelization – reduction to a kernel

Observation : we just proved that in polynomial time, we can

- decide whether an instance is negative ($n > k^2$),
- or compte an équivalent instance of size (polynomially) bounded by a fonction of k.

The problem ($\textsc{Point-Line Cover}, k$) admits a (quadratic) kernel

Kernelization – reduction to a kernel

Observation : we just proved that in polynomial time, we can

- decide whether an instance is negative ($n > k^2$),
- or compte an équivalent instance of size (polynomially) bounded by a fonction of k.

The problem ($\textsc{Point-Line Cover}, k$) admits a (quadratic) kernel

A kernelization for a parameterized problem is a polynomial time algorithm, that given an instance (G, k) returns an instance (G', k') such that:

- (G, k) is a positive instance $\Leftrightarrow$ (G', k') is a positive instance
- $|G'| \leqslant h(k)$ for some function $h : \mathbb{N} \to \mathbb{N}$
- $k' \leqslant k$

Existence of a kernel and fixed parameterized tractability

Theorem : A parameterized problem is **FPT** iff it is decidable and admits a kernelization.

Proof

$\Rightarrow$ Let $\mathcal{K}$ be a kernelization. Consider the following algorithm $\mathcal{A}$

1. compute $G' = \mathcal{K}(G)$ in time polynomial in $|G|$,
2. decide if $G' \in Q$ with an exact exponentiel algorithm $\mathcal{A}'$.

$\Leftarrow$ As $|G'| \leqslant h(\kappa(k))$, algorithm $\mathcal{A}$ runs in **FPT**–time.

$\Rightarrow$ Let $\mathcal{A}$ be a **FPT** algorithm with time complexity $f(k) \cdot n^c$ for some constant $c > 0$

- if $n = |G| \leqslant f(k)$, then the instance size is bounded,
- otherwise $f(k) \cdot n^c \leqslant n \cdot n^c = n^{c+1}$: thereby $\mathcal{A}$ runs in time polynomial in $|G|$.

Existence of a kernel and fixed parameterized tractability

Theorem : A parameterized problem is **FPT** iff it is decidable and admits a kernelization.

Proof

$\Rightarrow$ Let $\mathcal{K}$ be a kernelization. Consider the following algorithm $\mathcal{A}$

1. compute $G' = \mathcal{K}(G)$ in time polynomial in $|G|$,
2. decide if $G' \in Q$ with an exact exponentiel algorithm $\mathcal{A}'$.

$\Leftarrow$ As $|G'| \leqslant h(\kappa(k))$, algorithm $\mathcal{A}$ runs in **FPT**–time.

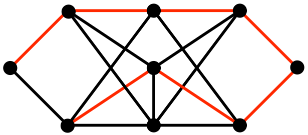
$\Rightarrow$ Let $\mathcal{A}$ be a **FPT** algorithm with time complexity $f(k) \cdot n^c$ for some constant $c > 0$

- if $n = |G| \leqslant f(k)$, then the instance size is bounded,
- otherwise $f(k) \cdot n^c \leqslant n \cdot n^c = n^{c+1}$: thereby $\mathcal{A}$ runs in time polynomial in $|G|$.

Observation : the resulting kernel size is exponential in k.

Does polynomial size kernel always exist ?

- A graph $G = (V, E)$ and a parameter $k \in \mathbb{N}$
- Does G contains a path of length k ?



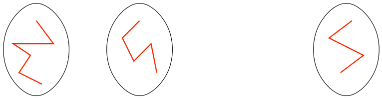
LONGEST PATH is **NP**–Complete (reduction to HAMILTONIAN PATH) but can be solved in time $O(c^k.n^{O(1)})$ with the COLOR CODING technique.

Does polynomial size kernel always exist ?

Hypothesis: there exists a kernelization $\mathcal{A}$ for Longest Path that computes a kernel of size $t = k^c$ bits.

- build an instance (G, k) from t distinct instances

$$(G, k) = (G_1, k) \oplus (G_2, k) \oplus \ \dots \ \oplus (G_t, k)$$

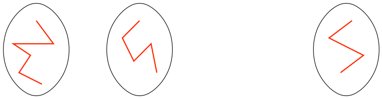


Does polynomial size kernel always exist ?

Hypothesis: there exists a kernelization $\mathcal{A}$ for LONGEST PATH that computes a kernel of size $t = k^c$ bits.

- build an instance (G, k) from t distinct instances

$$(G, k) = (G_1, k) \oplus (G_2, k) \oplus \quad \dots \quad \oplus (G_t, k)$$



Observation : (G, k) has a path of length k iff $\exists i$ st G_i as a path of length k.

Question :

Is it possible to decide whether one of the instances has a path of length k using less than 1 bit per instance in average ?

Non-existence of polynomial size kernel

Theorem: Unless co$-\mathbf{NP} \subseteq \mathbf{NP}/\text{poly}$, the parameterized LONGEST PATH problem does not admit a polynomial kernel.

Several tools exist to establish lower bounds on the kernel size (under some standard complexity assumptions)

- OR-composition [Bodlaender et al.] and AND-composition [Drucker]
- Polynomial and parameterized Transformations [Bodlaender et al.]
- Cross composition [Bodlaender et al.]
- ...

Parameterized intractability of **XP** problems

$\rightsquigarrow$ can we prove that *k*-Independent Set is not **FPT**?

Parameterized intractability of **XP** problems

$\rightsquigarrow$ can we prove that k-Independent Set is not **FPT**?

Exponential Time Hypothesis:

3-SAT cannot be solved in time $2^{o(n)}$

Theorem: If k-Clique or k-Independent Set can be solved in $f(k) \cdot n^{o(k)}$ time, then ETH is not valid.

Corollary: ETH $\Rightarrow$ k-Independent Set $\notin$ **FPT**

Parameterized intractability of **XP** problems

$\rightsquigarrow$ can we prove that k-Independent Set is not **FPT**?

Exponential Time Hypothesis:

3-SAT cannot be solved in time $2^{o(n)}$

Theorem: If k-Clique or k-Independent Set can be solved in $f(k) \cdot n^{o(k)}$ time, then ETH is not valid.

Corollary: ETH $\Rightarrow$ k-Independent Set $\notin$ **FPT**

We can define a hierarchy of complexity classes:

$$\mathbf{FPT} \subseteq \mathbf{W}[1] \subseteq \mathbf{W}[2] \subseteq \cdots \subseteq \mathbf{W}[t] \subseteq \cdots \subseteq \mathbf{W}[P] \subseteq \mathbf{XP}$$

Hypothèses :

- k-Clique and k-Independent Set are $W[1]$-complete
- k-Dominating Set is $W[2]$-complete

Synthesis

Under standard complexity hypothesis, we have observe that some **NP**–Complete problems:

- are **NP**–Complete for every fixed k
  k-Coloring **Para-NP-Complete**
- can be solved in time $O(n^k)$
  k-Independent Set **XP**
- can be solved in time $f(k) \cdot n^{O(1)}$
  k-Vertex Cover **FPT**
- does not admit a polynomial size kernel
  k-Longest Path **No-poly-Kernel**
- admit a polynomial size kernel
  k-Line-Cover **poly-Kernel**

Synthesis

Under standard complexity hypothesis, we have observe that some **NP**–Complete problems:

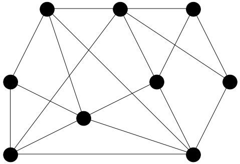
- are **NP**–Complete for every fixed k
  k-Coloring **Para-NP-Complete**
- can be solved in time $O(n^k)$
  k-Independent Set **XP**
- can be solved in time $f(k) \cdot n^{O(1)}$
  k-Vertex Cover **FPT**
- does not admit a polynomial size kernel
  k-Longest Path **No-poly-Kernel**
- admit a polynomial size kernel
  k-Line-Cover **poly-Kernel**

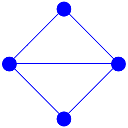
Observation: the Coloring problem is **FPT** time with respect to other parameters.

Part 2 – Meta-algorithmic theorems

An alternative story of parameterized algorithms

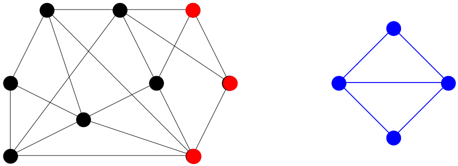
Graph Minors and Robertson & Seymour Theorem





H is a **minor** of G if it can be obtained by a series of

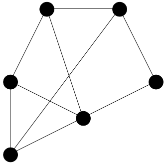
Graph Minors and Robertson & Seymour Theorem

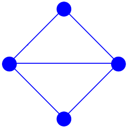


H is a **minor** of *G* if it can be obtained by a series of

- vertex deletions

Graph Minors and Robertson & Seymour Theorem

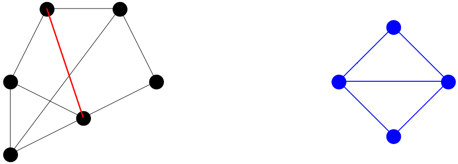




H is a **minor** of G if it can be obtained by a series of

- vertex deletions

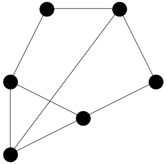
Graph Minors and Robertson & Seymour Theorem

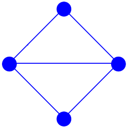


H is a **minor** of G if it can be obtained by a series of

- vertex deletions
- edge deletions

Graph Minors and Robertson & Seymour Theorem

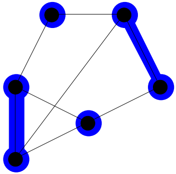


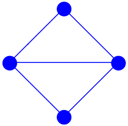


H is a **minor** of G if it can be obtained by a series of

- vertex deletions
- edge deletions

Graph Minors and Robertson & Seymour Theorem

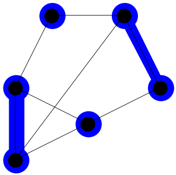


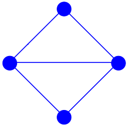


H is a **minor** of G if it can be obtained by a series of

- vertex deletions
- edge deletions
- edge contractions

Graph Minors and Robertson & Seymour Theorem





H is a **minor** of *G* if it can be obtained by a series of

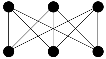
- vertex deletions
- edge deletions
- edge contractions

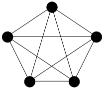
Theorem [Robertson & Seymour – Wagner's conjecture]
Graphs are well-quasi ordered by the minor relation.

Consequences of Robertson & Seymour Theorem

$\rightsquigarrow$ every graph family closed under minor is characterized by a finite list of forbidden minors.

- Planar graphs exclude

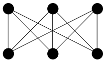


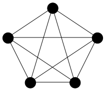


Consequences of Robertson & Seymour Theorem

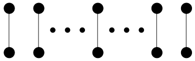
⇝ every graph family closed under minor is characterized by a finite list of forbidden minors.

- Planar graphs exclude



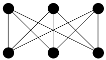
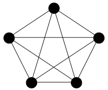
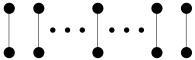


- Graphs with Vertex Cover of size $\leqslant k$ exclude



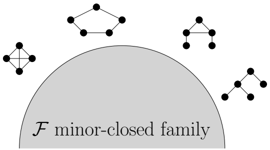
Consequences of Robertson & Seymour Theorem

$\rightsquigarrow$ every graph family closed under minor is characterized by a finite list of forbidden minors.

- Planar graphs exclude  
- Graphs with Vertex Cover of size $\leqslant k$ exclude 

Theorem [Robertson & Seymour]
In time $f(|H|) \cdot n^3$, we can test whether H is a minor of a graph G on n vertices.

Corollary
Every graph family closed under minor can be recognized in $O(n^3)$-time.



Consequences of Robertson & Seymour Theorem

By Robertson & Seymour Theorem and the minor inclusion test, we know that

Corollary

- k-Vertex Cover is (non-uniform) **FPT**.
- k-Feedback Vertex Set is (non-uniform) **FPT**.
- recognizing Treewidth $\leqslant k$ graphs is (non-uniform) **FPT**.
- recognizing graphs embeddable on a surface of genus k graphs is (non-uniform) **FPT**.
- ...

Consequences of Robertson & Seymour Theorem

By Robertson & Seymour Theorem and the minor inclusion test, we know that

Corollary

- k-Vertex Cover is (non-uniform) **FPT**.
- k-Feedback Vertex Set is (non-uniform) **FPT**.
- recognizing Treewidth $\leqslant k$ graphs is (non-uniform) **FPT**.
- recognizing graphs embeddable on a surface of genus k graphs is (non-uniform) **FPT**.
- ...

$\rightsquigarrow$ How to obtain constructive meta-algorithmic theorems ?

Courcelle's theorem

Theorem [Courcelle'91]
Let G be a graph and ϕ an $\mathbf{MSO}_2$ formula. Then deciding whether $G \models \phi$ is FPT with respect to parameter $\mathbf{tw}(G) + |\phi|$

- what is Monadic Second Order Logic ?
- what is **treewidth**?

Monadic Second Order Logic (on strings)

A string $w = x_1 \dots x_n \in \Sigma$ is represented by a structure $S(a)$ with universe $[n]$ equipped with

- a binary relation symbol $\leqslant$ representing the natural order on $[n]$;
- $\forall a \in \Sigma$, a unary relation symbol $P_a = \{i \in [n] \mid x_i = a\}$

Monadic Second Order Logic (on strings)

A string $w = x_1 \dots x_n \in \Sigma$ is represented by a structure $S(a)$ with universe $[n]$ equipped with

- a binary relation symbol $\leqslant$ representing the natural order on $[n]$;
- $\forall a \in \Sigma$, a unary relation symbol $P_a = \{i \in [n] \mid x_i = a\}$

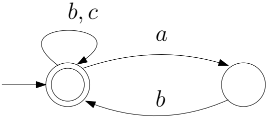
Example: $w = a \cdot b \cdot c \cdot c \cdot a \cdot b$

- $P_a = \{1, 5\}$ $\qquad$ $P_b = \{2, 6\}$ $\qquad$ $P_a = \{3, 4\}$

Monadic Second Order Logic (on strings)

A string $w = x_1 \ldots x_n \in \Sigma$ is represented by a structure $S(a)$ with universe $[n]$ equipped with

- a binary relation symbol $\leqslant$ representing the natural order on $[n]$;
- $\forall a \in \Sigma$, a unary relation symbol $P_a = \{i \in [n] \mid x_i = a\}$



$$\forall x \ (P_x a \rightarrow \exists y \ (x \leqslant y \wedge \forall z \ (z \leqslant x \vee y \leqslant z) \wedge P_y b))$$

Theorem [Büchi, McNaughton]
The following statements are equivalent

1. A language L on algphabet Σ is regular;
2. L can be recognized by a finite state automata;
3. there exists an $\mathbf{MSO}_2$ formula ϕ_L such that $w \in L$ iff $w \models \phi_L$.

Pumping lemma

Lemma [Scott, Rabin]

Let L be a regular language. There exists an integer $p \leqslant 1$ such that every word $w \in L$ of lenght $|w| \geqslant p$ can be written $w = x \cdot y \cdot z$ with

- $|y| \geqslant 1$ and $|x \cdot z| \leqslant p$ and $\forall i \in \mathbb{N}, x \cdot y^i \cdot z \in L$.

Pumping lemma

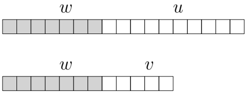
Lemma [Scott, Rabin]

Let L be a regular language. There exists an integer $p \leqslant 1$ such that every word $w \in L$ of lenght $|w| \geqslant p$ can be written $w = x \cdot y \cdot z$ with

- $|y| \geqslant 1$ and $|x \cdot z| \leqslant p$ and $\forall i \in \mathbb{N}, x \cdot y^i \cdot z \in L$.

Myhil-Nerode's equivalence classes: Let L be a regular language on alphabet Σ and let $u, v \in \Sigma^*$

$u \equiv_L v$ if $\forall w \in \Sigma^*, w \cdot u \in L \Leftrightarrow w \cdot v \in L$



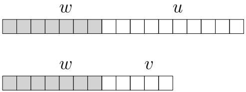
Pumping lemma

Lemma [Scott, Rabin]
Let L be a regular language. There exists an integer $p \leqslant 1$ such that every word $w \in L$ of lenght $|w| \geqslant p$ can be written $w = x \cdot y \cdot z$ with

- $|y| \geqslant 1$ and $|x \cdot z| \leqslant p$ and $\forall i \in \mathbb{N}, x \cdot y^i \cdot z \in L$.

Myhil-Nerode's equivalence classes: Let L be a regular language on alphabet Σ and let $u, v \in \Sigma^*$

$u \equiv_L v$ if $\forall w \in \Sigma^*, w \cdot u \in L \Leftrightarrow w \cdot v \in L$



- There exist finitely many equivalence classes
- Each equivalence class has a min-size representative u with $|u| \leqslant p$

Pumping lemma

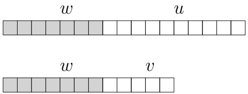
Lemma [Scott, Rabin]

Let L be a regular language. There exists an integer $p \leqslant 1$ such that every word $w \in L$ of lenght $|w| \geqslant p$ can be written $w = x \cdot y \cdot z$ with

- $|y| \geqslant 1$ and $|x \cdot z| \leqslant p$ and $\forall i \in \mathbb{N},\ x \cdot y^i \cdot z \in L$.

Myhil-Nerode's equivalence classes: Let L be a regular language on alphabet Σ and let $u, v \in \Sigma^*$

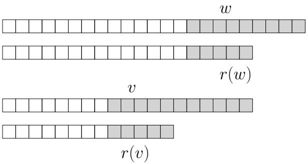
$u \equiv_L v$ if $\forall w \in \Sigma^*, w \cdot u \in L \Leftrightarrow w \cdot v \in L$



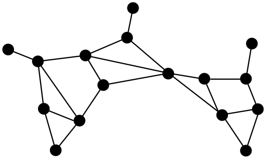
- There exist finitely many equivalence classes
- Each equivalence class has a min-size representative u with $|u| \leqslant p$

Algorithm to recognize L:

- Iteratively find a suffixe u, $p \leqslant |u| \leqslant 2p$
- Replace u by its representative $r(u)$.

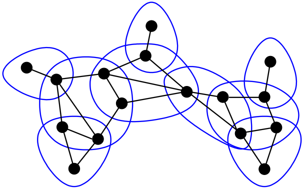


Generalization to trees and bounded treewidth graphs



A tree decomposition of a graph $G = (V, E)$ is a pair $(T, \{X_t : t \in T\})$ with T being a tree and $\forall t \in T$, $X_t \subseteq V$, such that

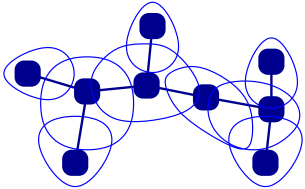
Generalization to trees and bounded treewidth graphs



A tree decomposition of a graph $G = (V, E)$ is a pair $(T, \{X_t : t \in T\})$ with T being a tree and $\forall t \in T$, $X_t \subseteq V$, such that

- [vertex covering] $\forall x \in V$, $\exists t \in T$ such that $x \in X_t$
- [edge covering] $\forall (x, y) \in E$, $\exists t \in T$ such that $x, y \in X_t$

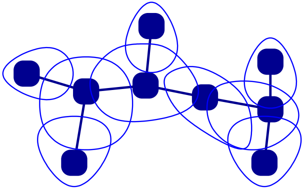
Generalization to trees and bounded treewidth graphs



A tree decomposition of a graph $G = (V, E)$ is a pair $(T, \{X_t : t \in T\})$ with T being a tree and $\forall t \in T$, $X_t \subseteq V$, such that

- [vertex covering] $\forall x \in V$, $\exists t \in T$ such that $x \in X_t$
- [edge covering] $\forall (x, y) \in E$, $\exists t \in T$ such that $x, y \in X_t$
- [consistency] $x \in V$ belongs to $X_{t_1} \cap X_{t_2}$, then $\forall t \in T$ on the $t_1 - t_2$-path in T, $x \in X_t$.

Generalization to trees and bounded treewidth graphs



A tree decomposition of a graph $G = (V, E)$ is a pair $(T, \{X_t : t \in T\})$ with T being a tree and $\forall t \in T$, $X_t \subseteq V$, such that

- [vertex covering] $\forall x \in V$, $\exists t \in T$ such that $x \in X_t$
- [edge covering] $\forall (x, y) \in E$, $\exists t \in T$ such that $x, y \in X_t$
- [consistency] $x \in V$ belongs to $X_{t_1} \cap X_{t_2}$, then $\forall t \in T$ on the $t_1 - t_2$-path in T, $x \in X_t$.

$$\mathbf{tw}(T, \mathcal{X}) = \max_{t \in V(T)} |X_t| - 1 \qquad \mathbf{tw}(G) = \min_{(T, \mathcal{X})} \mathbf{tw}(T, \mathcal{X})$$

$\mathbf{MSO}_2$ on graphs

A graph $w = x_1 \ldots x_n \in \Sigma$ is represented by a structure $S(a)$ with universe $V \cup E$ equipped with

- two unary relation symbols $\mathbf{V}$ and $\mathbf{E}$;
- a binary relation symbol $\mathbf{Inc} = \{(e, v) \mid \mathbf{E}e \wedge \mathbf{V}v \wedge v \in e\}$

$\mathbf{MSO}_2$ on graphs

A graph $w = x_1 \dots x_n \in \Sigma$ is represented by a structure $S(a)$ with universe $V \cup E$ equipped with

- two unary relation symbols $\mathbf{V}$ and $\mathbf{E}$;
- a binary relation symbol $\mathbf{Inc} = \{(e, v) \mid \mathbf{E}e \wedge \mathbf{V}v \wedge v \in e\}$

Example: G is a connected graph
$\forall V_1, V_2,$
$[\ \forall v \in V, (v \in V_1 \vee v \in V_2) \wedge (v \in V_1 \Rightarrow v \notin V_2) \wedge (v \in V_2 \Rightarrow v \notin V_1)\ ]$
$\wedge\ \ \exists v_1 \in V_1, \exists v_2 \in V_2, \exists e \in E, inc(v_1, e) \wedge inc(v_2, e)$

$\mathbf{MSO}_2$ on graphs

A graph $w = x_1 \ldots x_n \in \Sigma$ is represented by a structure $S(a)$ with universe $V \cup E$ equipped with

- two unary relation symbols **V** and **E**;
- a binary relation symbol $\mathbf{Inc} = \{(e, v) \mid \mathbf{E}e \wedge \mathbf{V}v \wedge v \in e\}$

Example: G is a connected graph

$$\forall V_1, V_2,$$
$$[\ \forall v \in V, (v \in V_1 \vee v \in V_2) \wedge (v \in V_1 \Rightarrow v \notin V_2) \wedge (v \in V_2 \Rightarrow v \notin V_1)\ ]$$
$$\wedge\ \ \exists v_1 \in V_1, \exists v_2 \in V_2, \exists e \in E, inc(v_1, e) \wedge inc(v_2, e)$$

Theorem [Courcelle'91]

Let G be a graph and ϕ an $\mathbf{MSO}_2$ formula. Then deciding whether

$G \models \phi$ is FPT with respect to parameter $\mathbf{tw}(G) + |\phi|$

Some consequences of Courcelle's theorem

Theorem

Vertex Cover parameterized by solution size is **FPT**

Proof

Some consequences of Courcelle's theorem

Theorem
VERTEX COVER parameterized by solution size is **FPT**

Proof

- if G has a VERTEX COVER of size at most k, then $\mathbf{tw}(G) \leqslant k + 1$
- $\mathbf{VC}(G) = \exists x_1, \ldots x_k \; (\mathbf{V}x_1 \ldots \mathbf{V}x_1 \; \wedge$
  $\forall e, \mathbf{E}e \; (\mathbf{Inc}(e, x_1) \vee \cdots \vee \mathbf{Inc}(e, x_k)))$

Some consequences of Courcelle's theorem

Theorem

Vertex Cover parameterized by solution size is **FPT**

Proof

- if G has a Vertex Cover of size at most k, then $\mathbf{tw}(G) \leqslant k + 1$
- $\mathbf{VC}(G) = \exists x_1, \ldots x_k \; (\mathbf{V}x_1 \ldots \mathbf{V}x_1 \; \wedge$
  $\forall e, \mathbf{E}e \; (\mathbf{Inc}(e, x_1) \vee \cdots \vee \mathbf{Inc}(e, x_k)))$

This applies to many other problems

- Feedback Vertex Set
- Dominating Set in planar graphs
- Graph Coloring in bounded treewidth graphs
- ...

Some consequences of Courcelle's theorem

Theorem

Vertex Cover parameterized by solution size is **FPT**

Proof

- if G has a Vertex Cover of size at most k, then $\mathbf{tw}(G) \leqslant k+1$
- $\mathbf{VC}(G) = \exists x_1, \dots x_k \ (\mathbf{V}x_1 \dots \mathbf{V}x_1 \ \wedge$
  $\forall e, \mathbf{E}e \ (\mathbf{Inc}(e, x_1) \vee \dots \vee \mathbf{Inc}(e, x_k)))$

This applies to many other problems

- Feedback Vertex Set
- Dominating Set in planar graphs
- Graph Coloring in bounded treewidth graphs
- ...

A serious drawback: Inefficient algorithms due to high exponential dependency in the parameter

Efficient meta-algorithms

$\mathcal{F}$-DELETION Problem

Given a graph $G = (V, E)$ and an integer k as parameter,

- is there a subset $X \subseteq V$ such that $G - X$ is $\mathcal{F}$-minor free ?

Efficient meta-algorithms

$\mathcal{F}$-Deletion Problem
Given a graph $G = (V, E)$ and an integer k as parameter,

- is there a subset $X \subseteq V$ such that $G - X$ is $\mathcal{F}$-minor free ?

Observations :

1. $\{K_2\}$-Deletion $\equiv$ Vertex Cover
$\equiv$ Treewidth-zero Vertex Deletion

2. $\{K_3\}$-Deletion $\equiv$ Feedback Vertex Set
$\equiv$ Treewidth-one Vertex Deletion

More generally, how fast can we solve

- Treewidth-t Vertex Deletion ?

Efficient meta-algorithms

$\mathcal{F}$-DELETION Problem

Given a graph $G = (V, E)$ and an integer k as parameter,

- is there a subset $X \subseteq V$ such that $G - X$ is $\mathcal{F}$-minor free ?

Observations :

1. $\{K_2\}$-DELETION $\equiv$ VERTEX COVER
   $\equiv$ TREEWIDTH-ZERO VERTEX DELETION
2. $\{K_3\}$-DELETION $\equiv$ FEEDBACK VERTEX SET
   $\equiv$ TREEWIDTH-ONE VERTEX DELETION

More generally, how fast can we solve

- TREEWIDTH-t VERTEX DELETION ?
- PLANAR-$\mathcal{F}$-DELETION ($\mathcal{F}$ contains a planar graph) ?

Known results (1)

When $\mathcal{F}$ is "non-planar"

- $\mathcal{F}$-Deletion is FPT
  (by the Roberston and Seymour' graph minor theorem)
- $\{K_5, K_{3,3}\}$-Deletion can be solved in $O^*(2^{2^{(k \log k)}})$
  [Marx, Schlotter'07] [Kawarabayashi'09]

When $\mathcal{F}$ is planar

- $\{K_2\}$-Deletion (VC) $O^*(1.2738^k)$ [J. Chen et al.'10]
- $\{K_3\}$-Deletion (FVS) $O^*(3.83^k)$ [Y. Cao et al',10],
- $\{\theta_c\}$-Deletion $O^*(c^k)$ [G. Joret et al.'11]
- $\{K_4\}$-Deletion $O^*(c^k)$ [E.J.Kim et al.'12]

Known results (2)

When $\mathcal{F}$ is planar (cont'd)

- $2^{2^{O(k \log k)}} \cdot n^{O(1)}$ -time algorithm based on DP
- $2^{O(k \log k)} \cdot n^2$ -time algorithm [Fomin et al.'11]
- $2^{O(k)} \cdot n \log^2 n$ -time algorithm for Connected-Planar-$\mathcal{F}$-Deletion [Fomin et al.'12]

Known results (2)

When $\mathcal{F}$ is planar (cont'd)

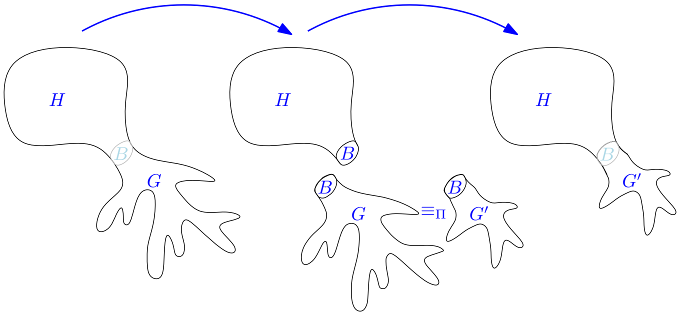
- $2^{2^{O(k \log k)}} \cdot n^{O(1)}$ -time algorithm based on DP
- $2^{O(k \log k)} \cdot n^2$ -time algorithm [Fomin et al.'11]
- $2^{O(k)} \cdot n \log^2 n$ -time algorithm for CONNECTED-PLANAR-$\mathcal{F}$-DELETION [Fomin et al.'12]

[KIM, LANGER, P., REIDL, ROSSMANITH, SAU, SIKDAR 2013]:

A $2^{O(k)}.n^2$ -time algorithm for PLANAR-$\mathcal{F}$-DELETION

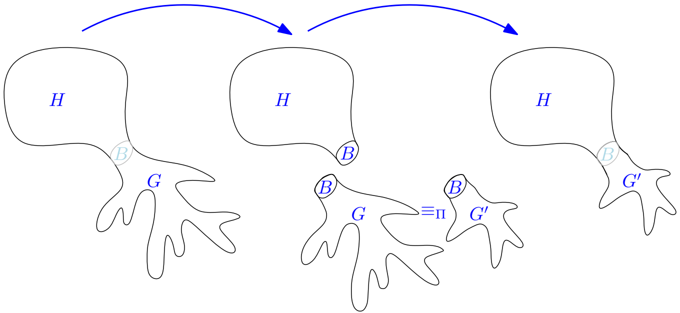
- [Chen et al.'05] No hope for a $2^{o(k)} \cdot n^{O(1)}$-time algorithm

Protrusion replacer



$$(G \oplus H, k) \Leftrightarrow (G' \oplus H, k + \Delta)$$

Protrusion replacer

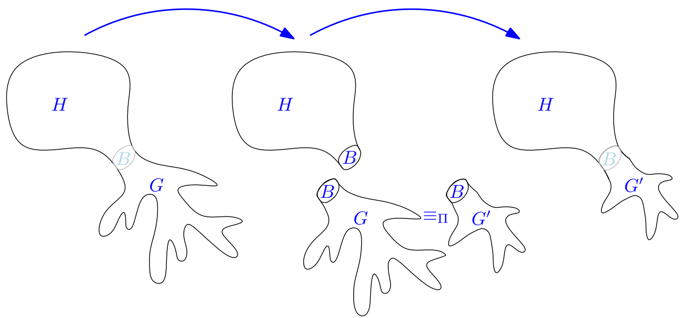


Let Π be a parameterized problem, G, G' be t-boundaried graphs we say that

$$G \equiv_{\Pi,t} G' \text{ if } \exists\ \Delta_\Pi(G, G') \text{ st } \forall\ H,$$

$$\boxed{(G \oplus H, k) \in \Pi \Leftrightarrow (G' \oplus H, k + \Delta_{\Pi,t}(G, G')) \in \Pi}$$

Protrusion replacer



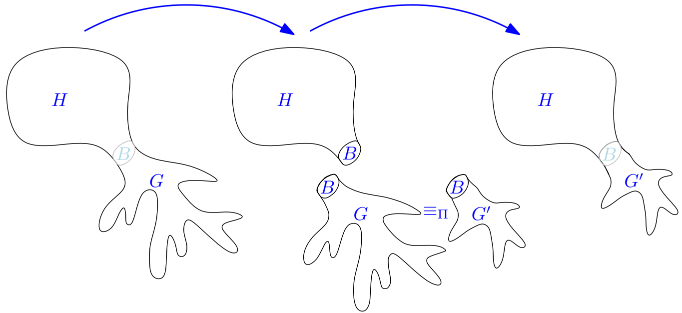
Let Π be a parameterized problem, G, G' be t-boundaried graphs we say that

$$G \equiv_{\Pi,t} G' \text{ if } \exists\ \Delta_\Pi(G, G') \text{ st } \forall\ H,$$

$$\boxed{(G \oplus H, k) \in \Pi \Leftrightarrow (G' \oplus H, k + \Delta_{\Pi,t}(G, G')) \in \Pi}$$

- Π has Finite Integer Index (FII) if $\equiv_{\Pi,t}$ has finitely many equivalence classes.

Protrusion replacer



Observation If Π has Finite Integer Index, then

replace G by one among finitely many representatives

- these representatives exists (depend only of Π and t)
- how large can they be ? how to compute them ?

Meta-algorithmic theorems for kernelization

- planar graphs — distance property

$\rightsquigarrow$ Alber, Fellows, Niedermeier — JACM'04
Polynomial-Time Data Reduction for Dominating Set

$\rightsquigarrow$ Guo, Niedermeier — ICALP'07
Linear problems kernels for NP-hard problems on planar graphs

- bounded genus — quasi-compactness

$\rightsquigarrow$ Bodlaender, Fomin, Lokshtanov, Penninkx, Saurabh, Thilikos
(Meta) Kernelization — FOCS'09

- H-minor free — bidim + separation ppty

$\rightsquigarrow$ Fomin, Lokshtanov, Saurabh, Thilikos — SODA'10
Bidimensionality and kernels

- topological minor free — treewidth-bounding

$\rightsquigarrow$ Kim, Langer, Paul, Reidl, Rossmanith, Sau, Sikdar — ICALP'13
Linear kernels [. . .] via protrusion decompositions

Meta-kernelization

Theorem [Bodlaender et al, FOCS'09]
If $\Pi \subseteq \mathcal{G}_g \times \mathbb{N}$ is a problem on graphs embedded in a surface st

- Π has Finite Integer Index and Π or $\bar{\Pi}$ is quasi-compact

then Π admits a linear kernel

Meta-kernelization

Theorem [Bodlaender et al, FOCS'09]
If $\Pi \subseteq \mathcal{G}_g \times \mathbb{N}$ is a problem on graphs embedded in a surface st

- Π has Finite Integer Index and Π or $\bar{\Pi}$ is quasi-compact

then Π admits a linear kernel

The proof is based on

- protrusion decomposition $\rightsquigarrow$ Can be computed in some cases
- protrusion replacer and MSO expressibility $\rightsquigarrow$ Existential only

Explicit kernel via Dynamic Programming

Theorem [Garnero, P, Sau, Thilikos'14]

Let (G, k) be an instance of a parameterized problem Π. Given

- given a protrusion decomposition of G and
- a Π-encoder $\mathcal{E}$ s.t. $\sim_{\mathcal{E}}$ is DP-friendly

we can construct a linear kernel for Π (with explicit constants)

Explicit kernel via Dynamic Programming

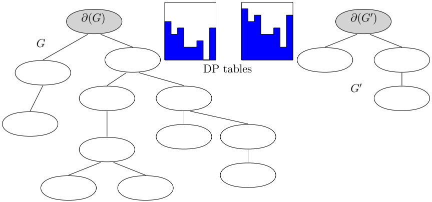
Theorem [Garnero, P, Sau, Thilikos'14]

Let (G, k) be an instance of a parameterized problem Π. Given

- given a protrusion decomposition of G and
- a Π-encoder $\mathcal{E}$ s.t. $\sim_{\mathcal{E}}$ is DP-friendly

we can construct a linear kernel for Π (with explicit constants)

t-boundaried graph $\rightsquigarrow$ rooted-tree decomposition (of width t) with root bag being the boundary.



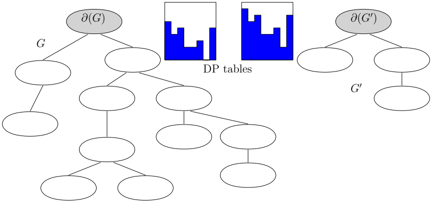
Explicit kernel via Dynamic Programming

Theorem [Garnero, P, Sau, Thilikos'14]
Let (G, k) be an instance of a parameterized problem Π. Given

- given a protrusion decomposition of G and
- a Π-encoder $\mathcal{E}$ s.t. $\sim_{\mathcal{E}}$ is DP-friendly

we can construct a linear kernel for Π (with explicit constants)

t-boundaried graph $\rightsquigarrow$ rooted-tree decomposition (of width t) with root bag being the boundary.



- Equivalence between boundaried graphs certified by DP-tables
- size of the DP-tables $\rightsquigarrow$ number of equivalence classes and
  $\rightsquigarrow$ size of the representative

Applications of the theorem

(treewidth modulator by [Fomin et al, SODA'10] +
protrusion decomposition by [Kim et al, ICALP'13])

- on graphs excluding an apex minor H
  - r-Dominating Set
  - r-Scattered Set

Applications of the theorem

(treewidth modulator by [Fomin et al, SODA'10] +
protrusion decomposition by [Kim et al, ICALP'13])

- on graphs excluding an apex minor H
  - r-Dominating Set
  - r-Scattered Set
- on graphs excluding a minor H
  - Connected Planar $\mathcal{F}$-Deletion
    (on topological minor free: uniform randomized kernel or deterministic but non-uniform wrt $\mathcal{F}$ and H - treewidth modulator by [Fomin et al, FOCS'12])
  - Connected Planar $\mathcal{F}$-Packing

(Connectedness is important to prove that $\sim_{\mathcal{E},g,t}$ is DP-friendly)

Applications of the theorem

(treewidth modulator by [Fomin et al, SODA'10] +
protrusion decomposition by [Kim et al, ICALP'13])

- on graphs excluding an apex minor H
  - r-Dominating Set
  - r-Scattered Set
- on graphs excluding a minor H
  - Connected Planar $\mathcal{F}$-Deletion
    (on topological minor free: uniform randomized kernel or deterministic but non-uniform wrt $\mathcal{F}$ and H - treewidth modulator by [Fomin et al, FOCS'12])
  - Connected Planar $\mathcal{F}$-Packing

(Connectedness is important to prove that $\sim_{\mathcal{E},g,t}$ is DP-friendly)

$\rightsquigarrow$ Dependency in the meta-parameter r or $r(\mathcal{F})$ is triple exponential

$$2^{2^{2^{c_H \cdot r \log r}}}$$

Thanks