KERNELIZATION

Sebastian Ordyniak

ALGORITHMS AND COMPLEXITY GROUP

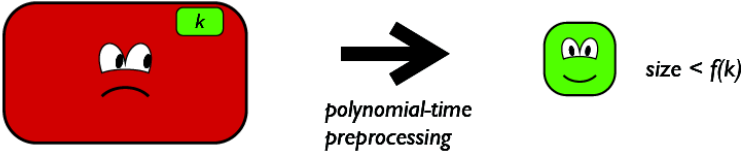
Introduction

- central topic of parameterized complexity with its own annual workshop **WORKER**, whose 2011 installment was organized in Vienna,
- polynomial-time preprocessing procedure with guarantees on the size of the reduced instance,
- close connection to approximation algorithms.

Motivation

- It is almost always a good idea to simplify the input.
- Preprocessing is successful in practice for SAT, CPLEX, TSP, etc..
- How to measure theoretically how well the preprocessing works?
- Performance guarantees?

Definition



- polynomial-time preprocessing for parameterized problems,
- the size of the reduced instance is bounded in the parameter,
- every problem in FPT allows such a reduction and vice versa.

Vertex Cover

k-Vertex Cover (*k*-VC) **Parameter:** k

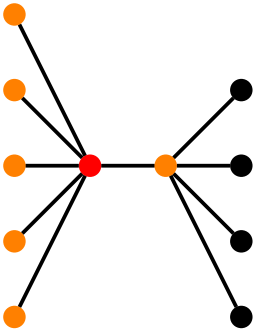
Input: A graph G and a natural number k.

Question: Does G have a vertex cover of size at most k, i.e. a set of vertices that covers all edges of G?

Vertex Cover Observation

Observation

For every vertex either itself or all of its neighbors occur in any vertex cover.



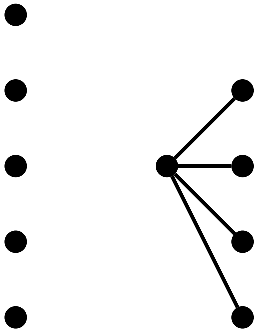
Vertex Cover Observation

Observation

For every vertex either itself or all of its neighbors occur in any vertex cover.

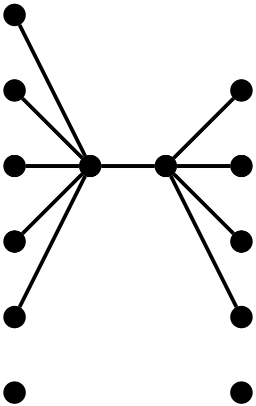
Rule 2

Take vertices of degree greater than k into the vertex cover (decrease k by 1 and delete them from the graph).



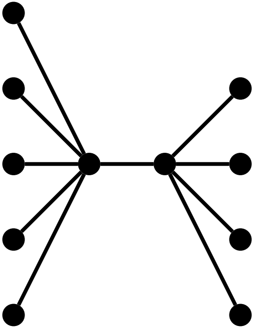
Vertex Cover Example

Does G have a vertex cover of size at most $k = 5$?



Vertex Cover Example

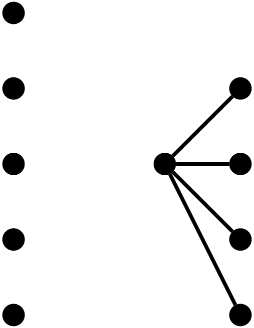
Does G have a vertex cover of size at most $k = 5$?



Remove isolated vertices

Vertex Cover Example

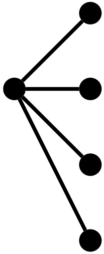
Does G have a vertex cover of size at most $k = 4$?



Take vertex of degree greater than $k = 5$ into the vertex cover, decrease k by one, and remove the vertex (Rule 2)

VERTEX COVER Example

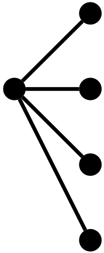
Does G have a vertex cover of size at most $k = 4$?



Remove isolated vertices

Vertex Cover Example

Does G have a vertex cover of size at most $k = 4$?



Now every vertex has degree at most $k = 4$!

The Kernel

Theorem

k-Vertex Cover has a kernel of size $O(k^2)$.

The Kernel

Theorem

k-VERTEX COVER has a kernel of size $O(k^2)$.

Remark:

The above theorem is easily extended to an FPT-algorithm:

- Compute the kernel in polynomial-time.
- Use brute-force on the kernel in time $O(2^{k^2})$.

Equivalence between FPT-algorithms and Kernelization

Theorem

A parameterized problem P is fixed-parameter tractable iff it is decidable and admits a kernelization algorithm.

Equivalence between FPT-algorithms and Kernelization

Theorem

A parameterized problem P is fixed-parameter tractable iff it is decidable and admits a kernelization algorithm.

Proof ($\rightarrow$)

- Assume P can be solved in time $f(k)n^{O(1)}$,
- if $n < f(k)$ then the instance is already a kernel,
- otherwise, i.e. if $n > f(k)$ then $f(k)n^{O(1)} \in n^{O(1)}$ and hence the instance can be solved in polynomial-time

Equivalence between FPT-algorithms and Kernelization

Theorem

A parameterized problem P is fixed-parameter tractable iff it is decidable and admits a kernelization algorithm.

Proof ($\leftarrow$)

- we can solve the instance by first computing the kernel of size $f(k)$ in time $n^{O(1)}$ and then running the algorithm that decides P on the kernel in time $g(f(k))$,
- Hence, the instance can be solved in time $n^{O(1)} + g(f(k))$ and P is fixed-parameter tractable.

Kernelsize

- even though every problem in FPT has a kernel of size $f(k)$, f should be as small as possible in order for the preprocessing to be as effective as possible.
- smaller kernels can usually be obtained by "more involved" case distinctions,
- suprisingly, one can distinguish between problems with and without polynomial sized kernels

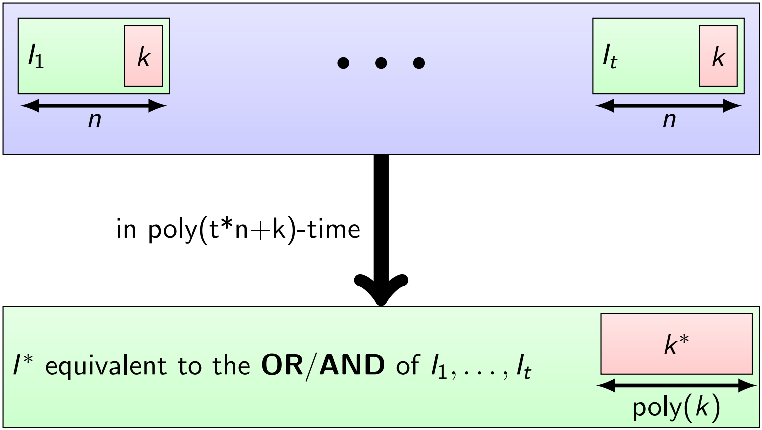
Lower Bounds

It is possible to show that a problem does not have a polynomial sized kernel [1] by either:

- providing a polynomial-parameter preserving fpt-reduction from a known problem without a polynomial kernel, or
- showing that the problem is OR/AND-composable

[1] Under the assumption that the polynomial hierachie does not collapse to its second level.

OR/AND-composition



OR-composition: Longest Path

k-LONGEST PATH (k-LP) **Parameter:** k

Input: A graph G and a natural number k.
Question: Does G have a path of length at least k?

OR-composition: Longest Path

k-LONGEST PATH (k-LP) **Parameter:** k

Input: A graph G and a natural number k.
Question: Does G have a path of length at least k?

Observation

Given t instances $(G_1, k), \cdots, (G_t, k)$ of LP, then:

$G_1 \dot{\cup} \cdots \dot{\cup} G_t$ is equivalent to the **OR** of $(G_1, k), \ldots, (G_t, k)$.

OR-composition: Longest Path

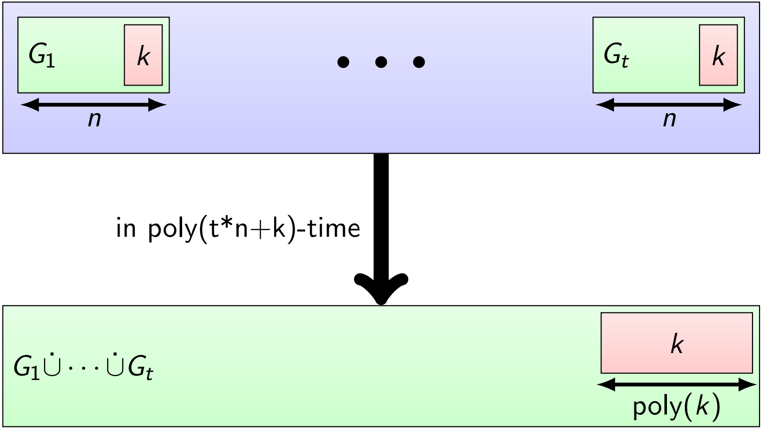


Table of Kernelization Results

Problem	**Kernel size**
Vertex Cover	$2k$
Connected Vertex Cover	no poly
Multiway Cut	?
Directed Multiway Cut	no poly
Almost-2-SAT	$O(k^6)$
Multicut	no poly
Pathwidth One Deletion Set	$O(k^2)$
Undirected Feedback Vertex Set	$4k^2$
Undirected Feedback Vertex Set	$4k^2$
Subset Feedback Vertex Set	?
Directed Feedback Vertex Set	?
Odd Cycle Transversal	$O(k^{4.5})$
Edge Bipartization	$O\ (k^3)$
Planar DS	$67k$
Max Leaf	$3.75k$
Directed Max Leaf	$O(k^2)$
Set Splitting	k
Nonblocker	$5k/3$
Edge Dominating Set	$2k^2 + 2k$
k-Path	no poly
Convex Recolouring	$O(k^2)$
Clique Cover	$2k$
Clique Partition	k^2
Cluster Editing	$2k$
Steiner Tree	no poly
3-Hitting Set	$O(k^2)$
Interval Completion	?
Minimum Fill-In	$2k^2 + 2k$
Contraction to Paths	$5k + 3$
Contraction to Trees	no poly

Meta-Kernelization

Parameterized by Solution Size

Various Meta-Kernelization results on "sparse classes of graphs" are known, e.g.:

- "compact" problems definable via **CMSO** admit a polynomial kernel on graph classes of bounded genus,
- "quasi compact" problems that have **FII** admit a linear kernel on graphs of bounded genus.
- ...

Meta-Kernelization

Parameterized by Structural Parameters

- Problems that have **FII** admit polynomial kernels parameterized by a treedepth-modulator on "sparse classes of graphs",
- Problems definable via **MSO** admit a linear (vertex-)kernel parameterized by the $\mathcal{C}$-cover number for any graph class $\mathcal{C}$ of constant rank-width.
- ...

Thank You!