Fixed-Parameter Algorithms, IA166

Sebastian Ordyniak

Faculty of Informatics
Masaryk University Brno

Spring Semester 2013

Tutorial – Topics

- The following is just a small selection of possible topics (papers) for the tutorial.
- Please ask for papers on topics you are interested in (you mind find out which topics these are during the curse of this lecture).
- If you know you are more interested in applications do not hesitate to ask as well!

Tutorial – Topics

- Jianer Chen, Iyad A. Kanj, Ge Xia: **Improved upper bounds for vertex cover.** Theor. Comput. Sci. 411(40-42): 3736-3756 (2010);
- Yixin Cao, Jianer Chen, Yang Liu: **On Feedback Vertex Set New Measure and New Structures.** SWAT 2010: 93-104;
- Jianer Chen, Yang Liu, Songjian Lu, Barry O'Sullivan, Igor Razgon: **A fixed-parameter algorithm for the directed feedback vertex set problem.** J. ACM 55(5): (2008);
- Dániel Marx, Igor Razgon: **Fixed-parameter tractability of multicut parameterized by the size of the cutset.** STOC 2011: 469-478;

Tutorial – Topics

- Jiong Guo, Jens Gramm, Falk Hüffner, Rolf Niedermeier, Sebastian Wernicke: **Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization**. J. Comput. Syst. Sci. 72(8): 1386-1396 (2006);
- Stefan Kratsch, Magnus Wahlström: **Compression via Matroids: A Randomized Polynomial Kernel for Odd Cycle Transversal**. SODA 2012: 94-103;
- Hans L. Bodlaender, Bart M. P. Jansen, Stefan Kratsch: **Kernel Bounds for Path and Cycle Problems**. IPEC 2011: 145-158;
- Hans L. Bodlaender, Bart M. P. Jansen, Stefan Kratsch:**Cross-Composition: A New Technique for Kernelization Lower Bounds**. STACS 2011: 165-176;

Tutorial – Topics

- Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, Dimitrios M. Thilikos: **Bidimensionality and Kernels.** SODA 2010: 503-510;
- Torben Hagerup: **Simpler Linear-Time Kernelization for Planar Dominating Set.** IPEC 2011: 181-193.

I will put a txt-file containing the titles of the papers on IS. If you want I can also upload the papers there!

Outline

Classical Complexity

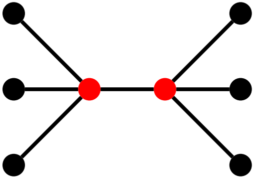
- The complexity of a problem is measured in terms of its **input size**.
- A problem is considered tractable if it can be solved in **polynomial time** and intractable if it is at least NP-hard.
- Unfortunately, many important problems are NP-hard so what can we do to tackle these problems?

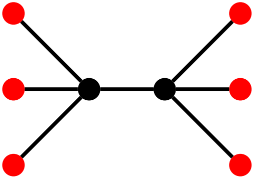
Parameterized Complexity

- The complexity of a problem is measured in terms of its **input size** and any number of additional **parameters.**
- Taking into account additional parameters:
    - provides a more fine-grained view of the complexity of a problem.
    - tells us where the exponential explosion of an NP-hard problem comes from.
    - allows us to design tailored algorithms for different parameterizations.

Parameterized Complexity

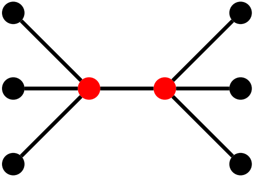
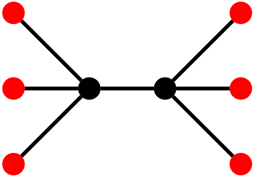
Problem:	MINIMUM VERTEX COVER	MAXIMUM INDEPENDENT SET
Input:	Graph G, integer k	Graph G, integer k
Question:	Is it possible to cover the edges with k vertices?	Is it possible to find k independent vertices?





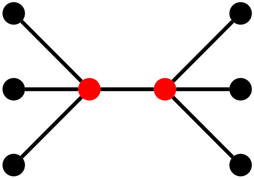
Classical	NP-complete	NP-complete

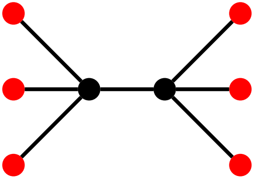
Parameterized Complexity

Problem:	MINIMUM VERTEX COVER	MAXIMUM INDEPENDENT SET
Input:	Graph G, integer k	Graph G, integer k
Question:	Is it possible to cover the edges with k vertices?	Is it possible to find k independent vertices?
		
Classical	NP-complete	NP-complete
trivial algorithm	$O(n^k)$	$O(n^k)$

Parameterized Complexity

Problem:	MINIMUM VERTEX COVER	MAXIMUM INDEPENDENT SET
Input:	Graph G, integer k	Graph G, integer k
Question:	Is it possible to cover the edges with k vertices?	Is it possible to find k independent vertices?





Classical	NP-complete	NP-complete
trivial algorithm	$O(n^k)$	$O(n^k)$
	$O(2^k n^2)$ algorithm exists	No $n^{o(k)}$ algorithm known

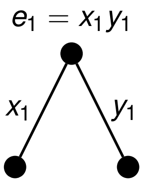
A simple algorithm for MINIMUM VERTEX COVER

Input: Graph G and integer k.

$$e_1 = x_1 y_1$$

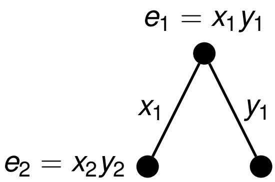
A simple algorithm for MINIMUM VERTEX COVER

Input: Graph G and integer k.



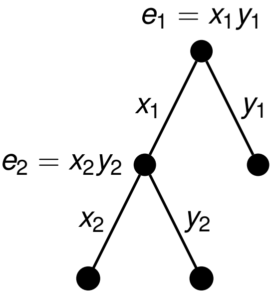
A simple algorithm for MINIMUM VERTEX COVER

Input: Graph G and integer k.



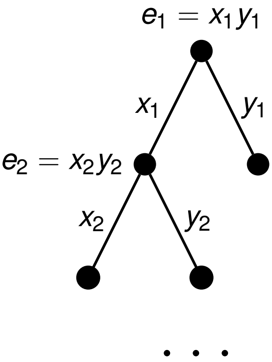
A simple algorithm for MINIMUM VERTEX COVER

Input: Graph G and integer k.



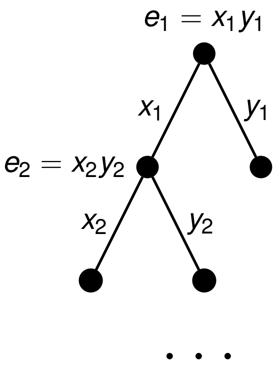
A simple algorithm for MINIMUM VERTEX COVER

Input: Graph G and integer k.



A simple algorithm for MINIMUM VERTEX COVER

Input: Graph G and integer k.



Running time

- at every node there are 2 choices;
- height of the search tree is at most k;
- number of nodes in the search tree is at most 2^k;
- complete search possible in $O(2^k n^c)$

Outline

Fixed-Parameter tractability

Definition

A **parameterization** of a decision problem is a function that assigns an integer parameter (usually denoted by k) to each input instance.

What can the parameter be?

- The size of the solution we are looking for.
- The maximum degree of the input graph.
- The diameter of the input graph.
- The length of clauses in the input SAT-formula.
- ...

Fixed-Parameter Tractability

Definition

A **parameterized problem** is a language $L \subseteq \Sigma^* \times \mathbb{N}$ where Σ is a finite alphabet. The first component is the classical input and second component is the **parameter** of the problem.

Definition

A parameterized problem is **fixed-parameter tractable (FPT)** if there is an $f(k)n^c$ time algorithm for an arbitrary function f of the parameter k, input size n, and some constant c.

Fixed-Parameter Tractability

Definition

A parameterized problem is **fixed-parameter tractable (FPT)** if there is an $f(k)n^c$ time algorithm for an arbitrary function f of the parameter k, input size n, and some constant c.

Example: MINIMUM VERTEX COVER parameterized by the solution size is FPT: we have already seen that it can be solved in time $O(2^k n^2)$.

Better algorithms are known (and are still being developed), e.g., $O(1.2832^k k + kn)$.

Main goal (of parameterized complexity): to find efficient FPT algorithms for NP-hard problems.

Fixed-Parameter Tractability

Definition

A parameterized problem is **fixed-parameter tractable (FPT)** if there is an $f(k)n^c$ time algorithm for an arbitrary function f of the parameter k, input size n, and some constant c.

Remarks

- O^*–notation: $O^*(f(k))$ means $O(f(k)n^c)$ for some constant c.
- unless otherwise stated we always use k to denote the parameter and n to denote the input size.

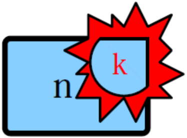
Fixed-Parameter Tractability

Examples of NP-hard problems that are FPT:

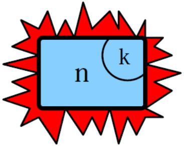
- Finding a vertex cover of size k.
- Finding a path of length k.
- Finding k disjoint triangles.
- Drawing the graph in the plane with k edge crossings.
- Finding disjoint paths that connect k pairs of points.
- ...

Fixed-Parameter Tractability

Fixed-parameter algorithms limit the exponential explosion to the parameter instead of the whole input size.



instead of



+ Guaranteed optimality of the solution.
+ Provable upper bounds on the computational complexity.
– Exponential running time.

Other approaches to tackle intractable problems:

- Randomized algorithms
- Approximation algorithms
- Heuristics
- Average Case Analysis
- New models of computing (DNA or quantum computing)

Outline

VERTEX COVER an illustrative example

VERTEX COVER — **Parameter:** k

Input: An undirected graph $G = (V, E)$ and a natural number k.
Question: Find a subset of vertices $C \subseteq V$ of size at most k such that each edge in E has at least one of its endpoints in C.

Solution methods:

- Bounded Search Tree: $O^*(1.28^k)$.
- Data reduction by preprocessing: techniques by Buss, Nemhauser Trotter.

VERTEX COVER an illustrative example

- Parameterizing
- Specializing
- Generalizing
- Counting or Enumeration
- Lower bounds
- Implementing and applying
- Exploiting the structure given by a VERTEX COVER for other problems

VERTEX COVER an illustrative example

- Parameterizing
  - Size of the vertex cover;
  - Dual parameterization: INDEPENDENT SET;
  - Parameterizing above guaranteed values, e.g., in planar graphs;
  - Structure of the input graph, e.g., treewidth
- Specializing
- Generalizing
- Counting or Enumeration
- Lower bounds
- Implementing and applying
- Exploiting the structure given by a VERTEX COVER for other problems

VERTEX COVER an illustrative example

- Parameterizing
- Specializing: special graph classes, e.g., planar graphs $O^*(c^{\sqrt{k}})$.
- Generalizing
- Counting or Enumeration
- Lower bounds
- Implementing and applying
- Exploiting the structure given by a VERTEX COVER for other problems

VERTEX COVER an illustrative example

- Parameterizing
- Specializing
- Generalizing: WEIGHTED VERTEX COVER, CAPACITATED VERTEX COVER, HITTING SET, ...
- Counting or Enumeration
- Lower bounds
- Implementing and applying
- Exploiting the structure given by a VERTEX COVER for other problems

VERTEX COVER an illustrative example

- Parameterizing
- Specializing
- Generalizing
- Counting or Enumeration:
  - Counting: $O^*(1.47^k)$.
  - Enumeration: $O^*(2^k)$.
- Lower bounds
- Implementing and applying
- Exploiting the structure given by a VERTEX COVER for other problems

Vertex Cover an illustrative example

- Parameterizing
- Specializing
- Generalizing
- Counting or Enumeration
- Lower bounds: widely open!
- Implementing and applying
- Exploiting the structure given by a Vertex Cover for other problems

VERTEX COVER an illustrative example

- Parameterizing
- Specializing
- Generalizing
- Counting or Enumeration
- Lower bounds
- Implementing and applying: re-engineering case distinctions, parallelization, . . .
- Exploiting the structure given by a VERTEX COVER for other problems

VERTEX COVER an illustrative example

- Parameterizing
- Specializing
- Generalizing
- Counting or Enumeration
- Lower bounds
- Implementing and applying
- Exploiting the structure given by a VERTEX COVER for other problems: solve related problems using an optimal vertex cover.

Outline

The Art of Problem Parameterization

- Parameter really small?
- Guaranteed parameter value?
- More than one obvious parameterization?
- Close to “trivial” problem instances?

Outline

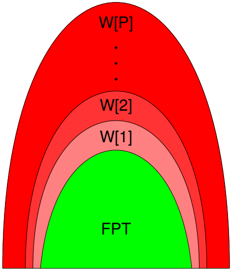
Algorithmic Techniques

Powerful toolbox for designing FPT algorithms with significant advances over the last 20 years:

Outline

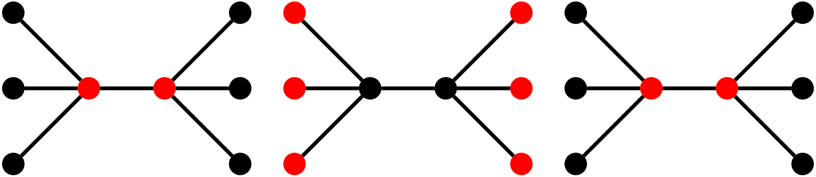
Fixed-Parameter Intractability

- Corresponding to the class NP in the classical setting in parameterized complexity there is a whole hierarchy of complexity classes (the **W-hierarchy**).
- All problems that are at least W[1]**-hard** are considered **fixed-parameter intractable**.
- Most natural problems are either FPT, W[1]-complete or W[2]-complete.



Parameterized Complexity Classes

Fixed-Parameter Intractability



Vertex Cover	Independent Set	Dominating Set
NP-complete	NP-complete	NP-complete
FPT	W[1]**-complete**	W[2]**-complete**

Literature

- **Rolf Niedermeier, Invitation to Fixed-Parameter Algorithms, Oxford University Press 2006**
- **Joerg Flum and Martin Grohe, Parameterized Complexity Theory, Springer 2006**
- Micheal R. Fellows and Rodney G. Downey, Parameterized Complexity, Springer 1999