Parameterized Complexity – An Overview

Serge Gaspers[1]

[1]CMM, University of Chile, Santiago, Chile

CC61X: Diseño y Analisis de Algoritmos Adaptivos

Outline

1. Introduction to Parameterized Complexity
   - Motivation
   - Examples of parameters
   - Main Complexity Classes
2. FPT Algorithms
   - Vertex Cover: definition and example
   - General Argument by Graph Minors
   - Algorithms for Vertex Cover
3. W-Hardness
   - Comparison: classical vs. parameterized hardness reduction
   - Induced Biclique: example and definition
   - Induced Biclique is $W[1]$-hard
4. Resources

Outline

A story

A computer scientist meets a biologist ...

$n = 1000$ experiments

$k = 20$ exp. failed

NP-hard

2-approx

$O(2^n)$ $\quad 2^{1000}$

$O(n^k)$ $\quad 1000^{20} \approx 10^{60}$

Eliminating conflicts from experiments

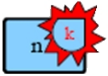
$n = 1000$ experiments
$k = 20$ experiments failed

	Running Time	
Theoretical	Nb of Instructions	Real
2^n	$1.07 \cdot 10^{301}$	$3.16 \cdot 10^{285}$ years
n^k	10^{60}	$2.95 \cdot 10^{44}$ years
$2^k \cdot n$	$1.05 \cdot 10^9$	0.97 seconds

Notes:
– We assume that 2^{30} instructions are carried out per second.
– The Big Bang happened roughly $13.5 \cdot 10^9$ years ago.

Goal of Parameterized Complexity

Confine the combinatorial explosion to a parameter k.



For which problem–parameter combinations can we find algorithms with running times of the form

$$f(k) \cdot n^{O(1)},$$

where the f is a computable function independent of the input size n?

Outline

1. Introduction to Parameterized Complexity
   - Motivation
   - Examples of parameters
   - Main Complexity Classes
2. FPT Algorithms
   - Vertex Cover: definition and example
   - General Argument by Graph Minors
   - Algorithms for Vertex Cover
3. W-Hardness
   - Comparison: classical vs. parameterized hardness reduction
   - Induced Biclique: example and definition
   - Induced Biclique is $W[1]$-hard
4. Resources

Examples of Parameters

A Parameterized Problem

- Input: an instance of the problem
- Parameter: a parameter
- Question: a Yes–No question about the instance and the parameter

- A parameter can be
  - solution size
  - input size (trivial parameterization)
  - related to the structure of the input (maximum degree, treewidth, branchwidth, genus, ...)
  - etc.

Outline

Main Complexity Classes

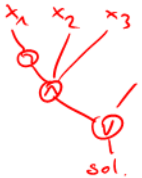
P: class of problems that can be solved in time $n^{O(1)}$
FPT: class of problems that can be solved in time $f(k) \cdot n^{O(1)}$
$W[\cdot]$: parameterized intractability classes

Vertex Cover Dominating Set

$$P \subseteq FPT \subseteq W[1] \subseteq W[2] \cdots \subseteq W[P]$$

Matching

Independent Set



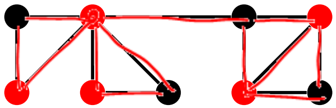
Known: If $FPT = W[1]$, then the Exponential Time Hypothesis fails, i.e. 3-SAT can be solved in time $O(2^{\epsilon \cdot n})$ for every $\epsilon > 0$.

Outline

Vertex Cover

VERTEX COVER (VC)

- Input: A graph $G = (V, E)$ on n vertices, an integer k.
- Parameter: k
- Question: Is there a set of vertices $C \subseteq V$ of size at most k such that every edge has at least one endpoint in C?



Outline

1. Introduction to Parameterized Complexity
   - Motivation
   - Examples of parameters
   - Main Complexity Classes
2. FPT Algorithms
   - Vertex Cover: definition and example
   - General Argument by Graph Minors
   - Algorithms for Vertex Cover
3. W-Hardness
   - Comparison: classical vs. parameterized hardness reduction
   - Induced Biclique: example and definition
   - Induced Biclique is $W[1]$-hard
4. Resources

General Argument by Graph Minors

Well-quasi-ordering: reflexive and transitive ordering which has no infinite antichain (any set of pairwise incomparable objects is finite)

$H \leq G \quad G \leq J \Rightarrow H \leq J \;//\; G \leq G$

Minor: A graph H is a minor of a graph G if H can be obtained by contracting edges of a subgraph of G.

By Robertson, Seymour:

- finite graphs are well quasi ordered under taking minors
- checking whether H is a minor of G can be done in time $f(|H|) \cdot |G|^{O(1)}$

$\rightarrow$ any graph problem whose Yes-instances (or No-instances) are closed under taking minors is FPT.

Graph Minor argument applied to VC

Yes-instances of VC are closed under taking minors.

Let G be a graph and C be a vertex cover of size $\leq k$ of G.

- if H is obtained from G by removing an edge, then C is a v.c. of size $\leq k$ of H
- if H is obtained from G by removing a vertex, ...
- if H is obtained from G by contracting an edge, ...

Thus, VC $\in$ FPT.
But: this argument does not give us an FPT algorithm, as the proofs are not constructive.

$2^{2^{2^{\cdot^{\cdot^{\cdot^{k}}}}}} \sim 2^{k}$

Outline

Introduction to Parameterized Complexity
Motivation
Examples of parameters
Main Complexity Classes
FPT Algorithms
Vertex Cover: definition and example
General Argument by Graph Minors
Algorithms for Vertex Cover
W-Hardness
Comparison: classical vs. parameterized hardness reduction
Induced Biclique: example and definition
Induced Biclique is W-hard
Resources

Brute Force Algorithms

$f(k) \cdot \text{poly}(n)$

2^n - algorithm
checking all subsets

- $2^n \cdot n^{O(1)}$ not FPT
- $n^k \cdot n^{O(1)}$ not FPT

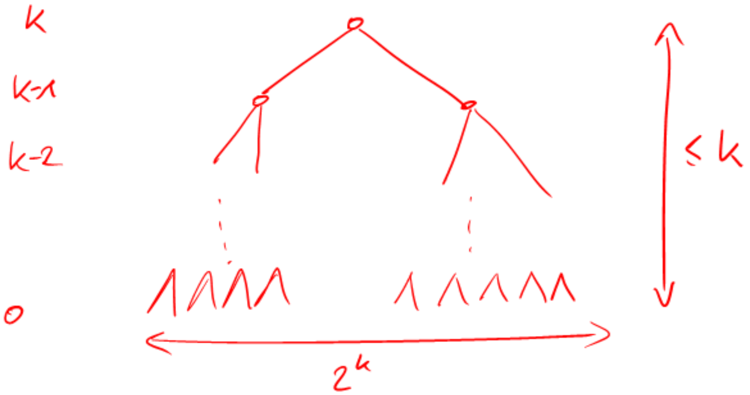
An FPT Algorithm

Algorithm vc1(G,k);
Input : A graph $G=(V,E)$, an integer $k \geq 0$
Output: Yes if G has a vertex cover of size $\leq k$,
No otherwise.

if $E=\emptyset$ **then** // all edges are covered
  return Yes
else if $k=0$ **then** // we used too many vertices
  return No
else
  Select an edge $uv \in E$;
  return vc1$(G-u,k-1) \vee$ vc1$(G-v,k-1)$

- Running time?

Running Time of Algorithm vc1



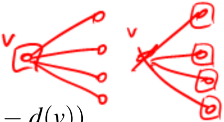
$\underline{\underline{2^k \cdot poly(n)}}$

can we decrease this?

A faster FPT Algorithm

Algorithm vc2(G, k);

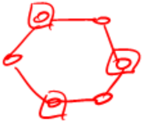
if $E = \emptyset$ **then** // all edges are covered
  return Yes
else if $k = 0$ **then** // we used too many vertices
  return No
else if $\Delta(G) \leq 2$ **then** // G has maximum degree ≤ 2
  Solve the problem in polynomial time;
else
  Select a vertex v of maximum degree;
  return vc2$(G - v, k - 1) \vee$ vc2$(G - N[v], k - d(v))$



- Running time?

Exercise

Show that VC can be solved in polynomial time for graphs of maxmium degree at most 2.



$P_n \rightarrow$ size of vertex cover is

$C_n \rightarrow$ size of vertex cover is

C is a v.c. containing u

$\rightarrow C' = (C \setminus \{u\}) \cup \{v\}$ is a v.c. of size $\leq |C|$

Running Time of Algorithm vc2

v has degree ≥ 3

$$T(k) \leq T(k-1) + T(k-3)$$

$$x^k = x^{k-1} + x^{k-3} \quad \Big| \cdot \frac{1}{x^{k-3}}$$

$$x^3 = x^2 + 1$$

$$x^3 - x^2 - 1 = 0$$

$\rightarrow$ solve using Newton's method

just one positive real root : $x \approx 1.4656\ldots$

$\rightarrow$ running time is $1.4656^k \cdot poly(n)$

$1.4656^k \cdot n^{O(1)}$

(Current best: $1.2738^k \cdot n^{O(1)}$)

Outline

Parameterized Reduction

Definition 1 (parameterized reduction)

A parameterized reduction is a transformation, taking as input an instance (I, k) of a problem Π_1, and producing an instance (I', k') of a problem Π_2 in time $f(k) \cdot |I|^{O(1)}$ such that

- (I, k) is a Yes-instance for $\Pi_1 \iff (I', k')$ is a Yes-instance for Π_2
- $k' \leq g(k)$ for a function g

$\Pi_1 \longrightarrow \Pi_2$

$(I, k) \longmapsto (I', k')$

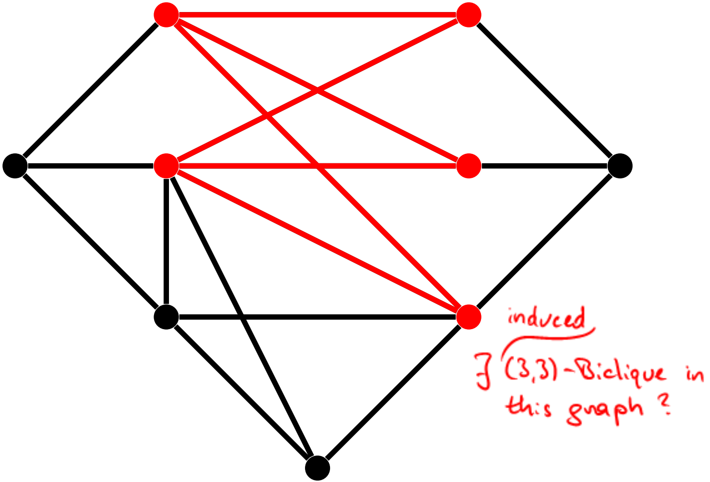
transformation done in time $f(k) \cdot |I|^{O(1)}$

YES $\iff$ YES

$k' \leq g(k)$

Outline

Definition by example



Induced (2,3)-Biclique $B \subseteq V$.

Definitions

solve IS. → use algo IB

Induced Biclique (IB)

- Input: $G = (V, E)$, k.
- Parameter: k
- Question: Does G have an induced (k, k)-Biclique?

Independent Set (IS)

- Input: $G = (V, E)$, k.
- Parameter: k
- Question: Does G have k pairwise non-adjacent vertices?

IS is a typical $W[1]$-complete problem.

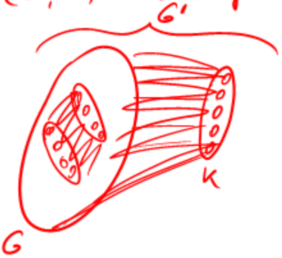
Outline

Independent Set reduces to Induced Biclique

(G,k) instance for ind. set.

$\downarrow$

(G',k') instance for induced Biclique



k-ind. set in G $\Rightarrow$ ind. (k,k)-Biclique in G'

ind. (k,k)-Biclique in G' $\Rightarrow$ k-ind. set in G

Ind. Set reduces to Induced Biclique by a parameterized Reduction.

Exercise

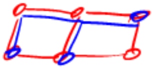
k-Leaf Tree

In: $G=(V,E)$, integer k

Param: k

Qu: Does G have a subtree with $\geq k$ leaves

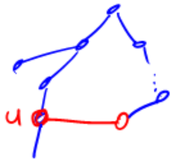
connected acyclic subgraph



$k=3$

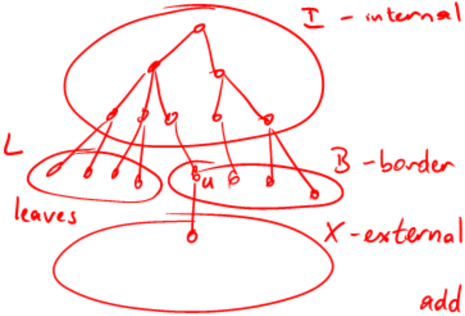
L. $u \in V$. $\exists$ opt. sol. T st. either u is a leaf in T or T contains all edges incident to u

Proof



$\square$

Task: Design a $4^k \cdot poly(n)$ time algorithm for k-Leaf Tree



- $|L| + |B| \geq k$: YES
- $|B| = 0$: No

Choose $u \in B$

① add u to L

② add u to I $\to$ put $N(u) \cap X$ into B

special case: $X \cap N(u) = \{v\}$

v leaf $\to$ exist equivalent sol. where u is a leaf

$\rightsquigarrow$ "follow path"

Run Time:

Potential fct:

$\mu = 2k - 2|L| - |B| \leq 2k$

① μ decreases by 1

② μ decreases by ≥ 1

Run time $2^{\mu} \leq 2^{2k} = 4^{k}$

Resources

- Books
  - R.G. Downey, M. R. Fellows: Parameterized Complexity. Springer-Verlag, 1999.
  - J. Flum and M. Grohe. Parameterized Complexity Theory. Springer-Verlag, 2006.
  - R. Niedermeier. Invitation to Fixed-Parameter Algorithms. Oxford University Press, 2006.
  - M. R. Fellows and F. A. Rosamond. Parameterized Complexity Extremal Theory. Cambridge University Press. In progress.
  - H. Fernau. Parameterized Algorithmics: A Graph-Theoretic Approach. Habilitationsschrift, Wilhelm-Schickard Institut für Informatik, Universität Tübingen, 2005. Book in progress.
- WIKI: `http://fpt.wikidot.com`