Fixed Parameterized Algorithms and Exact Exponential Algorithms

Evaluation – Mars 2012

Overall objective

Efficiently solve combinatorial problems, mainly in graph theory.

- Design as efficient as possible algorithms, from a theoretical point of view.
- Get a better understanding of the hardness of problems.
- Compare our algorithms to practice: implementation and experimentation.

The obstacle

Conjecture: $\mathbf{P} \neq \mathbf{NP}$.

Lots of problems (NP-hard) cannot be solved in a theoretically fast time, i.e. in polynomial time.

How to solve them? Several methods:

- approximation algorithms;
- randomized algorithms;
- heuristics;
- **exact algorithms.**

Exact algorithms

Exact exponential algorithms: running time in c^n with c as small as possible. $\Rightarrow$ if c is small, one can solve instances of important size.
Sometimes subexponential algorithms .

Fixed parameter algorithms : running time in $f(k)P(n)$, where

- k parameter (well chosen),
- f arbitrary function,
- P polynomial.

$\Rightarrow$ if k is small, one can solve.

Optimization problems and parameterization

NP minimization problem

Instance: $x \in \Sigma^*$ where Σ finite alphabet.

Goal: Find $\min\{cost(x, y) \mid y \in sol(x)\}$ with

- $sol(x)$ set of solutions of x;
- $cost : \{(x, y) \mid y \in sol(x)\} \to \mathbb{N}$.

Associated decision problem

Instance: $x \in \Sigma^*$ and integer k.

Question: $\min\{cost(x, y) \mid y \in sol(x)\} \leq k$?

Associated parameterized problem

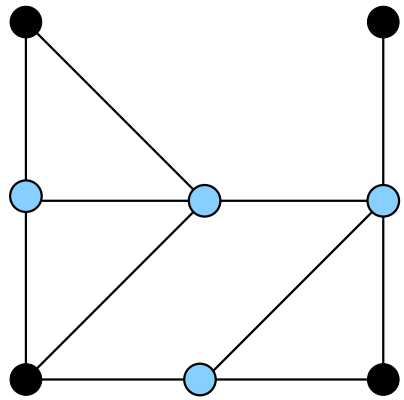
Instance: $x \in \Sigma^*$ and integer k.

Parameter: k.

Question: $\min\{cost(x, y) \mid y \in sol(x)\} \leq k$?

Example 1: Vertex cover

Vertex cover = set of vertices C s. t. every edge has an end in C.



Minimum Vertex Cover:

Instance: Graph G.

Goal: Find a minimum-size vertex cover G.

Parameterized Minimum Vertex Cover:

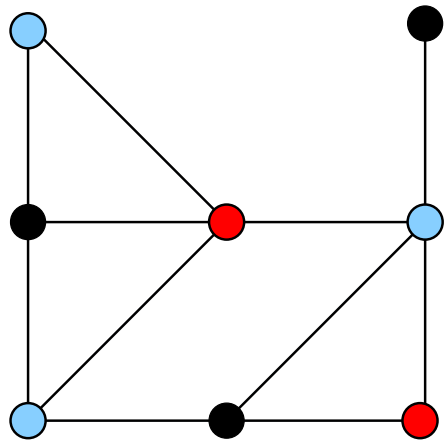
Instance: Graph G and integer k.

Parameter: k.

Question: does G have a vertex cover of size (at most) k ?

Example 2: Chromatic Number

colouring $= c$: $V(G) \to S$ s. t. $c(u) \neq c(v)$, $\forall uv \in E(G)$.



Chromatic Number:

Instance: Graph G.

Goal: Find a colouring of G with minimum number of colours.

Parameterized Colourability:

Instance: Graph G and integer k.

Parameter: k.

Question: is G k-colourable? ($\chi(G) \leq k$?)

FPT Problems

A parameterized problem is FPT (Fixed Parameter Tractable) if it can be solved in $f(k)n^c$ time.

FPT implies polynomial-time solvable for each **fixed** k $= XP$.

Min. Vertex Cover, Max. Stable Set, and Min. Dominating Set are XP.
Exhaustive algorithm testing all $O(n^k)$ k-subsets.

Colourability is not XP because 3-Colourability is NP-complete, so it is not FPT.

Eggemann, Havet et Noble. k-L(2,1)-Labelling for Planar Graphs is NP-Complete for $k \geq 4$. *Discrete Applied Math.* 158(16): 1777-1788, 2010

Parameterized Complexity

$$\mathsf{P} \subseteq \mathsf{FPT} \subseteq \underbrace{\mathsf{W[1]} \subseteq \mathsf{W[2]} \subseteq \ldots \subseteq \mathsf{W[P]}}_{\text{likely non FPT}} \subseteq \mathsf{XP}$$

Conjecture $\mathsf{FPT} \neq \mathsf{W[1]}$, and more generally $\mathsf{W[i]} \neq W[i+1]$

$$\mathsf{P} = \mathsf{NP} \Rightarrow \mathsf{FPT} = \mathsf{W[1]}$$

but the converse does not seem to be true.

Examples: Min. Vertex Cover is FPT, Max. Stable Set is W[1], Min. Dominating Set is W[2]


Amini, Sau, and Saurabh. Parameterized Complexity of the Smallest Degree-Constrained Subgraph Problem. *IWPEC 2008*.

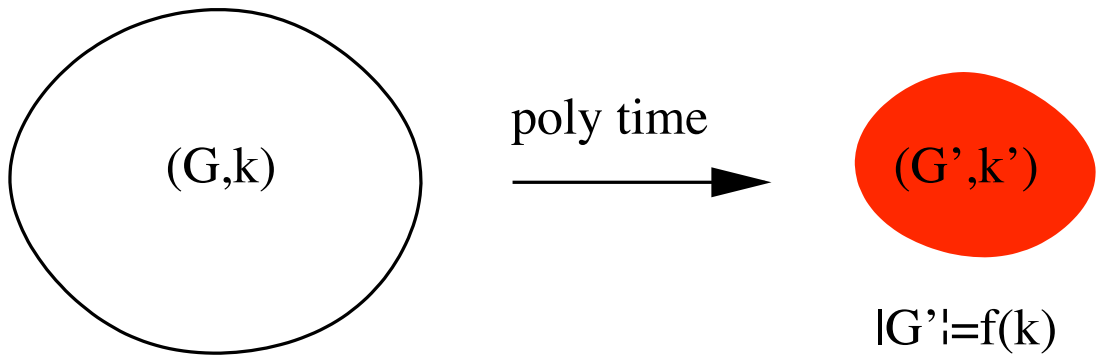
Techniques for FPT algorithms

- Reduction rules and kernels.
   Gonçalves, Havet, Pinlou and Thomassé. The spanning galaxy problem. *Discrete Applied Math.*, to appear.
- Bounded search tree
   Havet and Sampaio. On the Grundy number of a graph. *IPEC* 2010.
- Iterative compression
- Treewidth, minor theory.
- Color Coding
- . . .

Reduction to a kernel: Kernelization

$f(k)$-kernelization: polynomial-time algorithm
instance $(G, k) \longrightarrow$ instance (G', k') such that:

- (G', k') is equivalent to (G, k);
- $k' \leq k$ and $|G'| \leq f(k)$.



Kernelization + brute force = algorithm in time $O(g(k) + n^c)$.

Theorem: A parameterized problem is FPT if and only if it admits a kernelization.

Small kernels

We want small kernels, of polynomial size if possible.

Finding small kernels:

- Integer Linear Programming,
- Crown decomposition,
- ...

Non-existence of small kernels:

- Distillation: if a distillation algorithm exists, then there is no polynomial-size kernel (unless $PH = \Sigma_p^3$).
  $\rightarrow$ parameter preserving transformations in polynomial time.
   Guillemot, Havet, Paul and Perez. On the (non-)existence of polynomial kernels for P_l-free edge modification problems. *Algorithmica*, to appear.
- Reinforcement with colouring.

Perspectives

Develop and use existing methods.

Find new methods (probabilistic?, discharging-like?...)

Links between FPT and approximation algorithms. In particular, relation between kernels and approximations (better upper bounds / other lower bounds).

Envisaged problems:

- colouring. Grundy number FPT? Polynomial kernels of Dual Greedy Colouring and Dual b-Colouring?
  Meaningful parameter for which Colourability is FPT.
- digraphs. Subdivision of digraphs. Is k-linkage in acyclic digraphs FPT ?
  Parameterization with feedback vertex set.

Parameterized complexity problems on digraphs

Feedback vertex set (fvs), feedback arc set (fas), cycle packing number (cpn).

$$cpn \leq fvs \leq fas$$

.

	General digraphs	Tournaments
cpn $\leq k$?	W[1]	XP NP-complete ? FPT ?
fvs $\leq k$?	FPT Polynomial kernel ?	$O(k^3)$-kernel Smaller kernel ?
fas $\leq k$?	FPT Polynomial kernel ?	$(2+\epsilon)k$-kernel Smaller kernel ?

.

Exponential Algorithms

Classical methods.

- Dynamic programming.
- Branching algorithms.

   Bessy and Havet. Enumerating the edge-colourings and total colourings of a regular graph. *J. Combin. Optimization*, to appear.
- Sort and Search
- Inclusion-Exclusion

New methods.

- Running-time analysis of branching algorithms. Measure and Conquer, Branch and Recharge.

   Havet, Klazar, Kratochvil, Kratsch, and Liedloff. Exact algorithms for L(2,1)-labeling of graphs. *Algorithmica* 59(2):169–194, 2011.
- Tree decomposition.
- Iterative compression.

Perspectives

Develop and use existing methods.

Find new methods (probabilistic?, discharging-like?...)

Envisaged problems:

- colouring (and variants),
- counting and enumerating solutions
- restriction of some problems to graph classes.

Theoretical efficiency vs et practical efficiency

Courcelle's Theorem

If the treewidth is bounded ($\leq k$), then every problem expressable in MSOL logic is solvable in $f(k).n$ time.

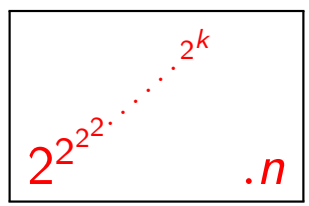
BUT what about the exact complexity? hidden constant

Height of the expression depends on the the number of quantifiers in the logical expression

Theoretical efficiency vs et practical efficiency

Courcelle's Theorem
If the treewidth is bounded ($\leq k$), then every problem expressable in MSOL logic is solvable in $f(k).n$ time.

BUT what about the exact complexity? hidden constant



Height of the expression depends on the the number of quantifiers in the logical expression

Measuring performances of an algorithm

Classical complexity does not totally reflect the behaviour of an algorithm.

- hidden constants.
- worst-case complexity.

Complementary analysis:

- Implementation et experimentation.
- Average complexity.

$\Longrightarrow$ sampling problem.
Notion of classes of hard instances $\rightarrow$ kernels.