9b. Review

COMP6741: Parameterized and Exact Computation

Serge Gaspers

School of Computer Science and Engineering, UNSW Sydney, Australia

19T3

Outline

Outline

1. Review
   - Upper Bounds
   - Lower Bounds

2. Research in Parameterized and Exact Computation

Outline

Kernelization: definition

Definition 1

A kernelization for a parameterized problem Π is a **polynomial time** algorithm, which, for any instance I of Π with parameter k, produces an **equivalent** instance I' of Π with parameter k' such that $|I'| \leq f(k)$ and $k' \leq f(k)$ for a computable function f.
We refer to the function f as the size of the kernel.

Main Complexity Classes

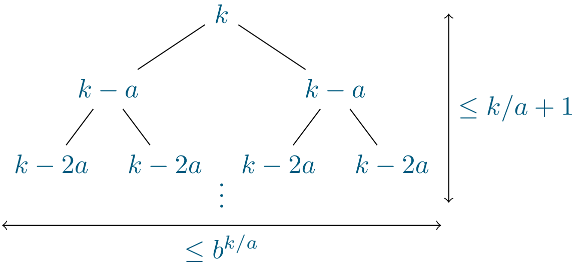
P: class of problems that can be solved in time $n^{O(1)}$
FPT: class of problems that can be solved in time $f(k) \cdot n^{O(1)}$
W$[\cdot]$: parameterized intractability classes
XP: class of problems that can be solved in time $f(k) \cdot n^{g(k)}$

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

Known: If FPT $=$ W$[1]$, then the Exponential Time Hypothesis fails, i.e. 3-SAT can be solved in time $2^{o(n)}$.

Search trees

Recall: A search tree models the recursive calls of an algorithm.
For a b-way branching where the parameter k decreases by a at each recursive call, the number of nodes is at most $b^{k/a} \cdot (k/a+1)$.



If k/a and b are upper bounded by a function of k, and the time spent at each node is FPT (typically, polynomial), then we get an FPT running time.

Measure & Conquer

Lemma 2 (Measure & Conquer Lemma)

Let

- A *be a branching algorithm*
- $c \geq 0$ *be a constant, and*
- $\mu(\cdot), \eta(\cdot)$ *be two measures for the instances of* A,

such that on input I, A *calls itself recursively on instances* $I_1, \ldots, I_k$, *but, besides the recursive calls, uses time* $O(|I|^c)$, *such that*

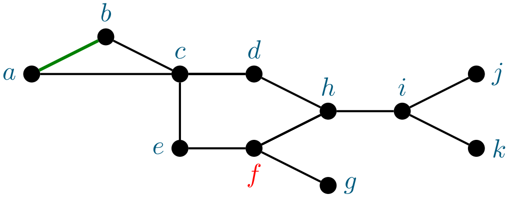
$$(\forall i) \quad \eta(I_i) \leq \eta(I) - 1, \text{ and} \tag{1}$$

$$2^{\mu(I_1)} + \ldots + 2^{\mu(I_k)} \leq 2^{\mu(I)}. \tag{2}$$

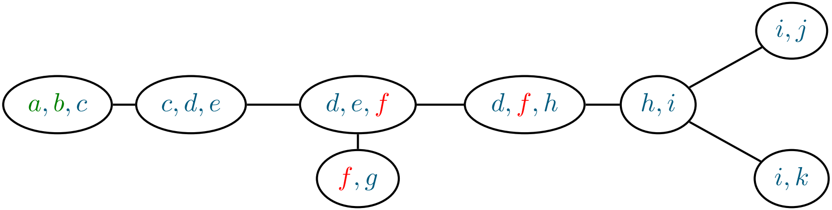
Then A *solves any instance* I *in time* $O(\eta(I)^{c+1}) \cdot 2^{\mu(I)}$.

Tree decompositions (by example)

- A graph G



- A *tree decomposition* of G



Conditions: covering and connectedness.

Randomized algorithms

Solution intersects a linear number of edges:

- Sampling vertices with probability proportional to their degree gives good success probability if the set of vertices we try to find has large intersection with the edges of the graph.

Color Coding:

Lemma 3

Let $X \subseteq U$ be a subset of size k of a ground set U.
Let $\chi : U \to \{1, \ldots, k\}$ be a random coloring of U.
The probability that the elements of X are colored with pairwise distinct colors is $\geq e^k$.

Monotone Local Search:

- For many subset problems a $O^*(c^k)$ algorithm for finding a solution of size k can be turned into a randomized algorithm finding an optimal solution in time $O^*((2-1/c)^n)$.

Outline

1. Review
   - Upper Bounds
   - Lower Bounds

2. Research in Parameterized and Exact Computation

Reductions

We have seen several reductions, which, for an instance (I, k) of a problem Π, produce an equivalent instance I' of a problem Π'.

	time	parameter	special features	used for
parameterized reduction	FPT	$k' \leq g(k)$		W[·]-hardness
polynomial parameter transformation	poly	$k' \leq \text{poly}(k)$		(Kernel LBs) (S)ETH LBs
SubExponential Reduction Family	subexp(k)	$k' \in O(k)$	Turing reduction $\|I'\| = \|I\|^{O(1)}$	ETH LBs

Outline

News

- Recently solved open problems from [DF13]
  - BICLIQUE is W[1]-hard [Lin18]
  - SHORT GENERALIZED HEX is W[1]-complete [Bon+17]
  - Determining the winner of a PARITY GAME is FPT in the number of values [Cal+17]
- research focii
  - enumeration algorithms and combinatorial bounds
  - randomized algorithms
  - treewidth: computation, bounds
  - bidimensionality
  - bottom-up: improving the quality of subroutines of heuristics
  - (S)ETH widely used now, also for poly-time lower bounds
  - FPT-approximation algorithms, lossy kernels
  - general-purpose "modeling" problems: SAT, CSP, ILP, Integer Quadratic Programming
  - backdoors

Resources

- FPT wiki: `http://fpt.wikidot.com`
- FPT newsletter: `http://fpt.wikidot.com/fpt-news:the-parameterized-complexity-newsletter`
- cstheory stackexchange: `http://cstheory.stackexchange.com`
- FPT summer schools (include lecture slides)
  - 2017: `https://algo2017.ac.tuwien.ac.at/pcss/`
  - 2014: `http://fptschool.mimuw.edu.pl`
  - 2009: `http://www-sop.inria.fr/mascotte/seminaires/AGAPE/`
- The Parameterized Algorithms and Computational Experiments Challenge (PACE): `https://pacechallenge.wordpress.com/`

References I

- [Bon+17] Édouard Bonnet, Serge Gaspers, Antonin Lambilliotte, Stefan Rümmele, and Abdallah Saffidine. "The Parameterized Complexity of Positional Games". In: *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*. Vol. 80. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017, 90:1–90:14.
- [Cal+17] Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. "Deciding parity games in quasipolynomial time". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2017)*. ACM, 2017, pp. 252–263.
- [DF13] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.
- [Lin18] Bingkai Lin. "The Parameterized Complexity of the k-Biclique Problem". In: *Journal of the ACM* 65.5 (2018), 34:1–34:23.