Parameterized Complexity

Maria Serna

Fall 2025

Parameterized complexity

- Possibility to give evidence that certain problems are not FPT.
- Parameterized reduction.
- The W-hierarchy.
- Tighter lower bounds.
- Relation to approximation
- Kernel size lower bounds
- Metatheorems

FPT-reductions

- Let (L, κ) and (L', κ') be two parameterized problems (on the same alphabet Σ)
- A FPT-reduction from (L, κ) to (L', κ') is a mapping $R : \Sigma^* \rightarrow \Sigma^*$ where
  - $\forall x \in \Sigma^*$ $x \in L$ iff $R(x) \in L'$
  - There is an FPT-algorithm with respect to κ computing R (in $f(\kappa(x))p(|x|))$
  - There is a computable function $g : \mathbb{N} \rightarrow \mathbb{N}$ such that $\forall x \in \Sigma^* \kappa'(R(x)) \leq g(\kappa(x))$
- We note $(L, \kappa) \leq^{fpt} (L', \kappa')$ when there is a FPT-reduction from (L, κ) to (L', κ')

FPT-reductions

Lemma

FPT is closed under FPT-reductions

FPT-reductions and complexity classes

- FPT-equivalence
  $(L, \kappa) \equiv^{fpt} (L', \kappa')$: $(L, \kappa) \leq^{fpt} (L', \kappa')$ and $(L', \kappa') \leq^{fpt} (L, \kappa)$

- p-INDEPENDENT SET $\equiv^{fpt}$ p-CLIQUE

$$R(G, k) = (\overline{G}, k)$$

  Works for both directions

- p-HITTING SET $\equiv^{fpt}$ p-DOMINATING SET
  Exercise

FPT-reductions and complexity classes

- Closure under FPT-reductions
  $[(L,\kappa)]^{fpt} = \{(L',\kappa') \mid (L',\kappa') \leq^{fpt} (L,\kappa)\}$
- If $\mathcal{C}$ is a class of parameterized problems
  - (L,κ) is $\mathcal{C}$-hard if $\mathcal{C} \subseteq [(L,\kappa)]^{fpt}$.
  - (L,κ) is $\mathcal{C}$-complete if $(L,\kappa) \in \mathcal{C}$ and (L,κ) is $\mathcal{C}$-hard.
- $[(L,\kappa)]^{fpt}$ defines a class of parameterized problems for which (L,κ) is complete
- if (L,κ) is $\mathcal{C}$-complete and $\mathcal{C}$ is closed under FPT reductions, then $\mathcal{C} = [(L,\kappa)]^{fpt}$

FPT-equivalent problems

The class paraNP

- Let (L, κ) be a parameterized problem
- (L, κ) belongs to paraNP if there is a non-deterministic algorithm $\mathcal{A}$ that decides $x \in L$ in time $f(\kappa(x))p(|x|)$, for some computable function f and polynomial function p.
- If $L \in$ NP, for each parameterization κ, $(L, \kappa) \in$ paraNP
  p-Clique, p-Vertex Cover, ... belong to paraNP.

paraNP-completeness

- Let (L, κ) be a parameterized problem
- (L, κ) is trivial if $L = \emptyset$ or $L = \Sigma^*$.
- The *i*-th slice of (L, κ) is the decision problem $(L, \kappa)_i = \{x \in L \mid \kappa(x) = i\}$

Theorem

If $(L, \kappa) \in$ paraNP is not trivial and has a NP-complete slice, then (L, κ) is paraNP-complete under FPT reductions.

paraNP-completeness:problems

- p-Vertex Coloring is paraNP-complete.
- p-Clique is not paraNP-complete, unless P = NP.
- p#var-Sat is not paraNP-complete, unless P = NP.
- pMax#Lit-Sat is paraNP-complete.

- paraNP-completeness separates *all slices* in P from *a slice* is NP-hard.

The class XP

- Let (L, κ) be a parameterized problem.
- (L, κ) belongs to (uniform) XP if there is an algorithm $\mathcal{A}$ that decides L in time $O(|x|^{f(\kappa(x))})$, for some computable function f.

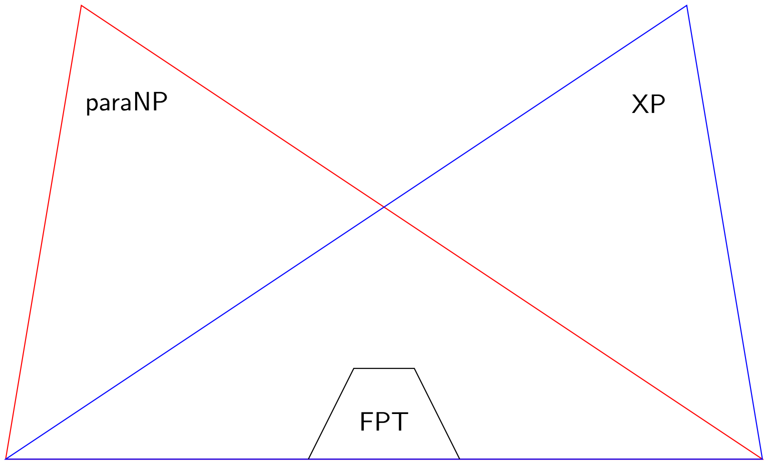
- p-Clique, p-Vertex Cover, p-Hitting Set, p-Hitting Set, p-Dominating Set belong to XP.
- XP is the counterpart of EXP in classic complexity.

XP-complete problems

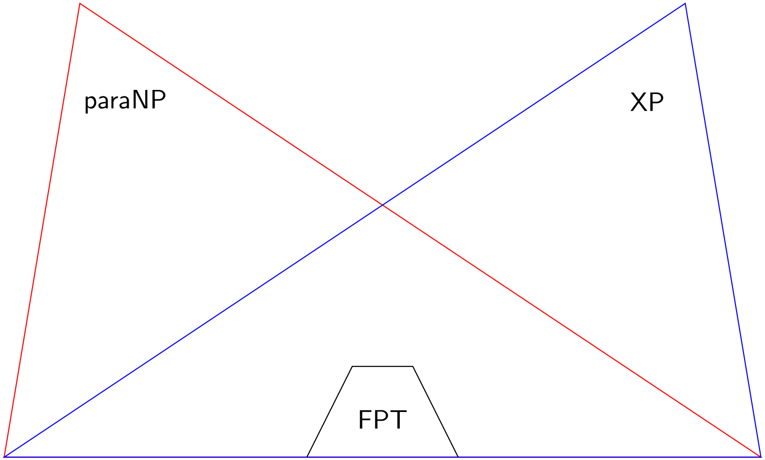
p-Exp-DTM-Halt
Input: A deterministic TM $\mathbb{M}$, $x \in \Sigma^*$ and an integer k,
Parameter: k
Question: Does $\mathbb{M}$ on input x stop in no more than $|x|^k$ steps?

Theorem
p-Exp-DTM-Halt *is XP-complete but does not belong to FPT unless $P = NP$.*

Relationships among classes



The W-hierarchy



Circuits: Depth and Weft

- Let C be a boolean circuit: `AND OR NOT` gates.
- A gate is small if it has only two or one input otherwise the gate is big
- The depth of C is the maximum distance from an input gate to an output gate.
- The weft of C the maximum number of big gates in a path from an input gate to an output gate.

- Note that $\text{depth}(C) \geq \text{weft}(C)$

Variations on SAT

- The weight of an assignment $x = x_1 \dots x_n \in \{0,1\}^n$ is $W(x) = \sum_{i=1}^{n} x_i$; i.e., the number of ones
- A circuit C is k-satisfiable if there is a satisfying assignment with weight k.
- A formula F is k-satisfiable if there is a satisfying assignment with weight k.

p-WSat(Fam)
Input: A circuit/formula C/F in family Fam and an integer k,
Parameter: k
Question: Is C/F k-satisfiable?

W-classes

Families of circuits/formulas

- $\textsc{Circ}$ all boolean circuits
- $\textsc{Prop}$ all propositional formulas
- For $d \geq t \geq 0$, define

$$\mathcal{C}_{t,d} = \{c \mid C \in \textsc{Circ} \text{ and } \mathrm{weft}(C) \leq t \text{ and } \mathrm{depth}(C) \leq d\}$$

We define the following classes:

- $W[P] = [\textsc{p-WSat}(\textsc{Circ})]^{fpt}$
- $W[SAT] = [\textsc{p-WSat}(\textsc{Prop})]^{fpt}$
- For $t \geq 1$, $W[t] = \{[\textsc{p-WSat}(C_{t,d})]^{fpt} \mid d \geq 1\}$

W-hierarchy

- $W[P] = [\textsc{p-WSat}(\textsc{Circ})]^{fpt}$
- $W[SAT] = [\textsc{p-WSat}(\textsc{Prop})]^{fpt}$
- For $t \geq 1$, $W[t] = \{[\textsc{p-WSat}(C_{t,d})]^{fpt} \mid d \geq 1\}$

Theorem

- *$W[P] \subseteq paraNP \cap XP$*
- *$W[SAT] \subseteq W[P]$*
- *For $i \geq 1$, $W[i] \subseteq W[SAT]$ and $W[i] \subseteq W[i+1]$*

W-hierarchy

Theorem
$FPT \subseteq W[1]$

Theorem

- *If, for some $i \geq 1$, $FPT \neq W[i]$ then $P \neq NP$*
- *If $FPT \neq W[SAT]$ then $P \neq NP$*
- *If $FPT \neq W[P]$ then $P \neq NP$*

Any of those conditions imply $FPT \neq paraNP$.

Theorem
If $FPT = W[P]$ then CIRCUITSAT *for circuits with n inputs and m gates can be decided in $2^{o(n)}m^{O(1)}$ time.*

W[P]-hard problems

Some problems in $W[P]$

- p-CLIQUE, p-DOMINANTSET, p-SETCOVER

But in which level of the W-hierarchy?

- p-CLIQUE $\in W[1]$
  To prove this statement it is enough to show a circuit with weft 1 solving the problem (see blackboard)
  In fact the problem is $W[1]$-complete
- p-DOMINATING SET $\in W[2]$ and p-SETCOVER $\in W[2]$ (Exercise)
  In fact both problems are $W[2]$-complete

Exponential Time Hypothesis

Exponential Time Hypothesis (ETH)

n-variable 3-SAT cannot be solved in time $2^{o(n)}$.

- We wish to get results like:
  If there is an $f(k)\ n^{o(k)}$ time algorithm for problem XXX, then ETH fails.

Lower bounds for FPT algorithms

- We know that VERTEX COVER can be solved in time $O^*(c^k)$.
- Can we do it much faster, for example in time $O^*(c^{\sqrt{k}})$ or $O^*(c^{k/logk})$?

Lemma
If VERTEX COVER can be solved in time $2^{o(k)}\ n^{O(1)}$, then ETH fails.

Proof.
There is a polynomial-time reduction from m-clause 3SAT to m-vertex VERTEX COVER. The assumed algorithm would solve the latter problem in time $2^{o(m)}\ n^{O(1)}$, violating ETH. □

Efficient approximation schemes

- Polynomial-time approximation scheme (PTAS):
  Input: Instance $x, \epsilon > 0$
  Output: $(1+\epsilon)$-approximate solution
  Running time: polynomial in $|x|$ for every fixed ϵ
- PTAS: running time is $|x|^{f(1/\epsilon)}$
- Efficient PTAS (EPTAS) running time is $f(1/\epsilon)|x|^{O(1)}$
- For some problems, there is a PTAS, but no EPTAS is known.

  Can we show that no EPTAS is possible?

No EPTAS?

Lemma

If the standard parameterization of an optimization problem is $W[1]$-hard, then there is no EPTAS for the optimization problem, unless $FPT = W[1]$.

Proof.

Suppose an $f(1/\epsilon)\ n^{O(1)}$ time EPTAS exists.
Running this EPTAS with $\epsilon = 1/(k+1)$ decides if the optimum is at most/at least k. □

Closure properties

- A contraction of an edge (u, v) in a graph G consists in replacing u, v by a new vertex w which keeps as neighbors $N(u) \cup N(v)$
- A graph H is a minor of G if H can be obtained from G by a series of edge contractions.
- If H is a minor of G then $tw(H) \leq tw(G)$

  (T, X) is a tree decomposition. Contract xy into z. The tree decomposition (T, X') in which we replace x, y by z in any bag containing x or z (or both) is a valid tree decomposition.

- If H is a subgraph of G then $tw(H) \leq tw(G)$

Algorithmic theorems

- Algorithmic Theorems provide a proof of the existence of an algorithm.
  - Vertex Cover, Dominating Set, 3-Coloring are solvable in linear time on graphs of constant treewidth.
  - Vertex Cover, Feedback Vertex Set can be solved in sub-exponential time on planar graphs
- To get an algorithm, as we have done, you should working out all the details!

Algorithmic meta theorems

- Algorithmic Meta Theorems. No algorithm is constructed!
- But the existence of an algorithm is proved
- Main uses: quick complexity classification tools, mapping the limits of applicability for specific techniques.
- Usually they are grounded in logics or other properties

First Order Logic on graphs

- We express graph properties using logic
- Basic vocabulary
  - Vertex variables: $x, y, z, \ldots$
  - Edge predicate $E(x, y)$, Equality $x = y$
  - Boolean connectives $\vee, \wedge, \neg$
  - Quantifiers $\forall$, $\exists$
- Example: Dominating Set of size 2

$$\exists x_1 \exists x_2 \forall y \; E(x_1, y) \vee E(x_2, y) \vee x_1 = y \vee x_2 = y$$

First Order Logic on graphs

- We express graph properties using logic
- Basic vocabulary
  - Vertex variables: $x, y, z, \ldots$
  - Edge predicate $E(x, y)$, Equality $x = y$
  - Boolean connectives $\vee, \wedge, \neg$
  - Quantifiers $\forall$, $\exists$
- Example: Vertex Cover of size 2

$$\exists x_1 \exists x_2 \forall y \forall z \; E(y, z) \rightarrow (y = x_1 \vee y = x_2 \vee z = x_1 \vee z = x_2$$

First Order Logic on graphs

- We express graph properties using logic
- Basic vocabulary
  - Vertex variables: $x, y, z, \ldots$
  - Edge predicate $E(x, y)$, Equality $x = y$
  - Boolean connectives $\vee, \wedge, \neg$
  - Quantifiers $\forall$, $\exists$
- Example: Clique of size 3

$$\exists x_1 \exists x_2 \exists x_3 \; E(x_1, x_2) \wedge E(x_1, x_3) \wedge E(x_2, x_3)$$

First Order Logic on graphs

- We express graph properties using logic
- Basic vocabulary
    - Vertex variables: $x, y, z, \ldots$
    - Edge predicate $E(x, y)$, Equality $x = y$
    - Boolean connectives $\vee, \wedge, \neg$
    - Quantifiers $\forall$, $\exists$
- Many standard (parameterized) problems can be expressed in FO logic.
- But some easy problems are inexpressible (e.g. connectivity).
- Rule of thumb: FO = local properties

Monadic Second Order Logic

- MSO logic: we add to FO logic
  - set variables $S_1, S_2, \ldots$
  - and the $a \in$ predicate.
  - Quantifiers $\forall$, $\exists$
    MSO$_1$ logic: we can quantify over sets of vertices only
    MSO$_2$ logic: we can quantify over sets of edges
- Example: 2-coloring

$$\exists V_1 \exists V_2 \forall x \forall y \; E(x,y) \to (x \in V_1 \leftrightarrow y \in V_2)$$

Algorithmic meta theorems

- All Monadic Second Order logic (MSO) expressible problems are solvable in linear time on graphs of constant treewidth.
- All minor closed optimization problems can be solved in sub-exponential time on planar graphs

Recall: The proof does not provide any algorithm!