



5

Parameterized Hardness

21 May 2025

Prof. Dr. Sebastian Wild

5 Parameterized Hardness

- 5.1 Parameterized Reductions
- 5.2 Nondeterministic FPT: Para-NP
- 5.3 Bounded Nondeterminism: $W[P]$
- 5.4 Tail-nondeterministic NRAM

How to prove \notin FPT?

► For some problems, no algorithm seems to achieve fpt running time

► example: p -CLIQUE

\rightsquigarrow maybe no fpt algorithm can exist for p -CLIQUE!

► **Problem:** Certainly exists in case $P = NP$

\rightsquigarrow strongest lower bound we can hope for will have to be conditional on $P \neq NP$

► Typical complexity-theory results:

No algorithm has a certain property unless some widely believed complexity hypothesis fails.

5.1 Parameterized Reductions

FPT Reductions

Definition 5.1 (Parameterized Reduction)

Let (L_1, κ_1) and (L_2, κ_2) be two parameterized problems (over alphabets Σ_1 resp. Σ_2).

An *fpt-reduction* (*fpt many-one reduction*) from (L_1, κ_1) to (L_2, κ_2) is a mapping $A : \Sigma_1^* \rightarrow \Sigma_2^*$ so that for all $x \in \Sigma_1^*$

1. (equivalence) $x \in L_1 \iff A(x) \in L_2$,
2. (fpt) A is computable by an fpt-algorithm (w.r.t. to κ_1), and
3. (parameter-preserving) $\kappa_2(A(x)) \leq g(\kappa_1(x))$ for a computable function $g : \mathbb{N} \rightarrow \mathbb{N}$.

We then write $(L_1, \kappa_1) \leq_{fpt} (L_2, \kappa_2)$.

Not all reductions are fpt

Many reductions from classical complexity theory are not parameter preserving.

Recall:

VERTEX-COVER

Given: graph $G = (V, E)$ and $k \in \mathbb{N}$

Question: $\exists V' \subset V : |V'| \leq k \wedge \forall \{u, v\} \in E : (u \in V' \vee v \in V')$

INDEPENDENT SET

Given: graph $G = (V, E)$ and $k \in \mathbb{N}$

Question: $\exists V' \subset V : |V'| \geq k \wedge \forall u, v \in V' : \{u, v\} \notin E$

5.2 Nondeterministic FPT: Para-NP

Parameterized NP: Non-deterministic NP

P corresponds to FPT ... but what is the analogue for NP?

Definition 5.2 (para-NP)

The class para-NP consists of all parameterized decision problems that are solved by a *non-deterministic* fpt-algorithm.

Some nice properties:

1. para-NP is closed under fpt-reductions.
2. $\text{FPT} = \text{para-NP} \iff \text{P} = \text{NP}$
3. an analogue for *kernalization* in FPT holds for para-NP

Can define para-NP-hard and para-NP-complete similarly as for NP:

Definition 5.3 (para-NP-hard)

(L, κ) is para-NP-hard if $(L', \kappa') \leq_{\text{fpt}} (L, \kappa)$ for all $(L', \kappa') \in \text{para-NP}$.

Coloring is para-NP-complete

As an example for a para-NP-complete problem, we consider graph coloring.

Theorem 5.4

p -GRAPHCOLORING is para-NP-complete.



Hello hardness, my old friend

Theorem 5.5 (para-NP-complete \rightarrow NP-complete for finite parameter)

Let (L, κ) be a nontrivial ($\emptyset \neq L \neq \Sigma^*$) parameterized problem that is para-NP-complete. Then $L_{\leq d} = \{x \in L : \kappa(x) \leq d\}$ is NP-hard.

The converse is essentially also true (see our proof for p -GRAPHCOLORING).

Proof:

para-NP-complete is too strict

Above Theorem means that many problems cannot be para-NP-complete!

For each of the following

- ▶ p -CLIQUE,
- ▶ p -INDEPENDENTSET
- ▶ p -DOMINATINGSET

bounding k by a **constant** d makes *polytime* algorithm possible.

↪ $L_{\leq d}$ cannot be NP-complete for each of these

- ▶ but we rather expect them \notin FPT

↪ para-NP theory does not settle complexity status

5.3 Bounded Nondeterminism: $W[P]$

Bye TM

para-NP is too large a class to have interesting complete problems

~> We must weaken the class. Unfortunately, TM inconvenient here.


Nondeterministic RAM

Definition 5.6 (NRAM, κ -restricted)


An NRAM M is a word-RAM with $w = O(\log n)$ with the additional operation to nondeterministically guess a number between 0 and a current register content.

An NRAM M that decides a parameterized problem (L, κ) is κ -restricted if on input $x \in \Sigma^*$ with $n = |x|$ and $k = \kappa(x)$

1. it performs at most $f(k) \cdot p(n)$ steps,
2. at most $g(k)$ of them nondeterministic,
3. uses at most $f(k) \cdot p(n)$ registers, and
4. those never contain numbers larger than $f(k) \cdot p(n)$.

for computable functions f and g . 

Definition 5.7 (W[P])

The class $\mathcal{W}[P]$ is the set of all parameterized problems (L, κ) decidable by a κ -restricted NRAM. 

A first $\mathcal{W}[P]$ -complete problem?

Define hardness and completeness for $\mathcal{W}[P]$ using \leq_{fpt} .

What could be the mother of all $\mathcal{W}[P]$ -complete problems?

Some parameterized version of SAT? Parameter #variables does not work. (Why?)

Weighted SAT

Definition 5.8 (Weighted Satisfiability)

Given: Boolean formula φ and integer $k \in \mathbb{N}$

Parameter: k

Question: \exists satisfying assignment with weight = k ?



Theorem 5.9 (p -WSAT(CIRC) is $\mathcal{W}[P]$ -complete)

The weighted satisfiability problem for boolean circuits parameterized by the weight is $\mathcal{W}[P]$ -complete.



5.4 Tail-nondeterministic NRAM

Tail-nondeterminism

Circuit satisfiability still too strong to show hardness of many interesting problems.

\rightsquigarrow We must weaken the class *further*.

Definition 5.10 (tail-nondeterministic NRAM)

A κ -restricted NRAM M for a problem (L, κ) is called *tail-nondeterministic* if all nondeterministic steps occur only among the last $h(\kappa(x))$ steps. ◀

Definition 5.11 ($\mathcal{W}[1]$)

The class $\mathcal{W}[1]$ consists of all parameterized decision problems (L, κ) that are decided by a tail-nondeterministic κ -restricted NRAM. ◀

As before, define hardness and completeness for $\mathcal{W}[1]$ w.r.t. \leq_{fpt} .

Stop

Definition 5.12 (k -step Halting Problem)

Given: A nondeterministic (single-tape) Turing machine M , an input x and $k \in \mathbb{N}$ be given.

Parameter: k

Question: Does M accepts x after at most k computation steps?

Remark 5.13

- ▶ M is part of input, so state space and tape alphabet are not fixed!
- \rightsquigarrow up to n different non-deterministic choices in *each* step. (n is size of encoding of M)
- \rightsquigarrow Trivial algorithm has to simulate up to n^{k+1} steps of M .
- ▶ Equivalent here to halting problem for $x = \varepsilon$, since we can hard-wire the given input into the states of a TM M' constructed from M .

W[1]-completeness

Theorem 5.14 (*k*-step halting problem $\mathcal{W}[1]$ -complete)

The *k*-step Halting Problem (for single-tape TM) parameterized by *k* is $\mathcal{W}[1]$ -complete. ◀

More natural problems?

Definition 5.15 (p -WSAT(2CNF))

Given: Boolean formula φ in 2-CNF and integer $k \in \mathbb{N}$

Parameter: k

Question: \exists satisfying assignment with weight = k ?



Theorem 5.16

p -WSAT(2CNF) is $\mathcal{W}[1]$ -complete.



Theorem 5.17

p -WSAT(2CNF⁻) is $\mathcal{W}[1]$ -complete.



p -WSAT(2CNF⁻) means: *all* literals *negated*.

p-Independent-Set is W[1]-complete

Theorem 5.18

p -INDEPENDENTSET is $\mathcal{W}[1]$ -complete.

Proof:



Partial Vertex Cover