W[1]-hardness

Dániel Marx

Recent Advances in Parameterized Complexity
Tel Aviv, Israel, December 3, 2017
Lower bounds

So far we have seen positive results: basic algorithmic techniques for fixed-parameter tractability.

What kind of negative results we have?

- Can we show that a problem (e.g., \textsc{Clique}) is \textbf{not} FPT?
- Can we show that a problem (e.g., \textsc{Vertex Cover}) has \textbf{no} algorithm with running time, say, $2^{o(k)} \cdot n^{O(1)}$?

This would require showing that $P \neq NP$: if $P = NP$, then, e.g., $k$-\textsc{Clique} is polynomial-time solvable, hence FPT.

Can we give some evidence for negative results?
Lower bounds

So far we have seen positive results: basic algorithmic techniques for fixed-parameter tractability.

What kind of negative results we have?

- Can we show that a problem (e.g., Clique) is not FPT?
- Can we show that a problem (e.g., Vertex Cover) has no algorithm with running time, say, $2^{o(k)} \cdot n^{O(1)}$?

This would require showing that $P \neq NP$: if $P = NP$, then, e.g., $k$-Clique is polynomial-time solvable, hence FPT.

Can we give some evidence for negative results?
Goals of this talk

Two goals:

1. Explain the theory behind parameterized intractability.
2. Show examples of parameterized reductions.
Classical complexity

**Nondeterministic Turing Machine (NTM):** single tape, finite alphabet, finite state, head can move left/right only one cell. In each step, the machine can branch into an arbitrary number of directions. Run is successful if at least one branch is successful.

**NP:** The class of all languages that can be recognized by a polynomial-time NTM.

**Polynomial-time reduction** from problem $P$ to problem $Q$: a function $\phi$ with the following properties:

- $\phi(x)$ is a yes-instance of $Q \iff x$ is a yes-instance of $P$,
- $\phi(x)$ can be computed in time $|x|^{O(1)}$.

**Definition:** Problem $Q$ is NP-hard if any problem in NP can be reduced to $Q$.

If an NP-hard problem can be solved in polynomial time, then every problem in NP can be solved in polynomial time (i.e., $P = NP$).
Parameterized complexity

To build a complexity theory for parameterized problems, we need two concepts:

- An appropriate notion of reduction.
- An appropriate hypothesis.

Polynomial-time reductions are not good for our purposes.

Example:

Graph $G$ has an independent set $k$ if and only if it has a vertex cover of size $n_k$.

Transforming an Independent Set instance $(G, k)$ into a Vertex Cover instance $(G, n_k)$ is a correct polynomial-time reduction.

However, Vertex Cover is FPT, but Independent Set is not known to be FPT.
Parameterized complexity

To build a complexity theory for parameterized problems, we need two concepts:

- An appropriate notion of reduction.
- An appropriate hypothesis.

Polynomial-time reductions are not good for our purposes.

**Example:** Graph $G$ has an independent set $k$ if and only if it has a vertex cover of size $n - k$.

$\Rightarrow$ Transforming an **Independent Set** instance $(G, k)$ into a **Vertex Cover** instance $(G, n - k)$ is a correct polynomial-time reduction.

However, **Vertex Cover** is FPT, but **Independent Set** is not known to be FPT.
Parameterized reduction

**Definition**

**Parameterized reduction** from problem $P$ to problem $Q$: a function $\phi$ with the following properties:

- $\phi(x)$ is a yes-instance of $Q$ $\iff$ $x$ is a yes-instance of $P$,
- $\phi(x)$ can be computed in time $f(k) \cdot |x|^{O(1)}$, where $k$ is the parameter of $x$,
- If $k$ is the parameter of $x$ and $k'$ is the parameter of $\phi(x)$, then $k' \leq g(k)$ for some function $g$.

**Fact:** If there is a parameterized reduction from problem $P$ to problem $Q$ and $Q$ is FPT, then $P$ is also FPT.
Parameterized reduction

**Definition**

Parameterized reduction from problem $P$ to problem $Q$: a function $\phi$ with the following properties:

- $\phi(x)$ is a yes-instance of $Q \iff x$ is a yes-instance of $P$,
- $\phi(x)$ can be computed in time $f(k) \cdot |x|^{O(1)}$, where $k$ is the parameter of $x$,
- If $k$ is the parameter of $x$ and $k'$ is the parameter of $\phi(x)$, then $k' \leq g(k)$ for some function $g$.

**Fact:** If there is a parameterized reduction from problem $P$ to problem $Q$ and $Q$ is FPT, then $P$ is also FPT.

**Non-example:** Transforming an Independent Set instance $(G, k)$ into a Vertex Cover instance $(G, n - k)$ is not a parameterized reduction.

**Example:** Transforming an Independent Set instance $(G, k)$ into a Clique instance $(\overline{G}, k)$ is a parameterized reduction.
**Multicolored Clique**

A useful variant of *Clique*:

**Multicolored Clique**: The vertices of the input graph $G$ are colored with $k$ colors and we have to find a clique containing one vertex from each color.

(or **Partitioned Clique**)

![Diagram of multicolored clique]

**Theorem**

There is a parameterized reduction from *Clique* to **Multicolored Clique**.
Multicolored Clique

**Theorem**

There is a parameterized reduction from **Clique** to **Multicolored Clique**.

Create $G'$ by replacing each vertex $v$ with $k$ vertices, one in each color class. If $u$ and $v$ are adjacent in the original graph, connect all copies of $u$ with all copies of $v$.

$k$-clique in $G$ $\iff$ multicolored $k$-clique in $G'$. 
**Multicolored Clique**

**Theorem**
There is a parameterized reduction from $\text{Clique}$ to $\text{Multicolored Clique}$.

Create $G'$ by replacing each vertex $v$ with $k$ vertices, one in each color class. If $u$ and $v$ are adjacent in the original graph, connect all copies of $u$ with all copies of $v$.

$k$-clique in $G \iff$ multicolored $k$-clique in $G'$.

*Similarly:* reduction to $\text{Multicolored Independent Set}$.
**Dominating Set**

**Theorem**

There is a parameterized reduction from **Multicolored Independent Set** to **Dominating Set**.

**Proof:** Let \( G \) be a graph with color classes \( V_1, \ldots, V_k \). We construct a graph \( H \) such that \( G \) has a multicolored \( k \)-clique iff \( H \) has a dominating set of size \( k \).

The dominating set has to contain one vertex from each of the \( k \) cliques \( V_1, \ldots, V_k \) to dominate every \( x_i \) and \( y_i \).
Theorem

There is a parameterized reduction from Multicolored Independent Set to Dominating Set.

Proof: Let $G$ be a graph with color classes $V_1, \ldots, V_k$. We construct a graph $H$ such that $G$ has a multicolored $k$-clique iff $H$ has a dominating set of size $k$.

- The dominating set has to contain one vertex from each of the $k$ cliques $V_1, \ldots, V_k$ to dominate every $x_i$ and $y_i$.
- For every edge $e = uv$, an additional vertex $w_e$ ensures that these selections describe an independent set.
Variants of **Dominating Set**

- **Dominating Set**: Given a graph, find $k$ vertices that dominate every vertex.
- **Red-Blue Dominating Set**: Given a bipartite graph, find $k$ vertices on the red side that dominate the blue side.
- **Set Cover**: Given a set system, find $k$ sets whose union covers the universe.
- **Hitting Set**: Given a set system, find $k$ elements that intersect every set in the system.

All of these problems are equivalent under parameterized reductions, hence at least as hard as **Clique**.
Basic hypotheses

It seems that parameterized complexity theory cannot be built on assuming $P \neq NP$ – we have to assume something stronger.

Let us choose a basic hypothesis:

**Engineers’ Hypothesis**

$k$-CLIQUE cannot be solved in time $f(k) \cdot n^{O(1)}$. 
Basic hypotheses

It seems that parameterized complexity theory cannot be built on assuming $P \neq NP$ – we have to assume something stronger.

Let us choose a basic hypothesis:

**Engineers’ Hypothesis**

$k$-Clique cannot be solved in time $f(k) \cdot n^{O(1)}$.

**Theorists’ Hypothesis**

$k$-Step Halting Problem (is there a path of the given NTM that stops in $k$ steps?) cannot be solved in time $f(k) \cdot n^{O(1)}$.
Basic hypotheses

It seems that parameterized complexity theory cannot be built on assuming $P \neq NP$ – we have to assume something stronger. Let us choose a basic hypothesis:

**Engineers’ Hypothesis**

$k$-Clique cannot be solved in time $f(k) \cdot n^{O(1)}$.

**Theorists’ Hypothesis**

$k$-Step Halting Problem (is there a path of the given NTM that stops in $k$ steps?) cannot be solved in time $f(k) \cdot n^{O(1)}$.

**Exponential Time Hypothesis (ETH)**

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

Which hypothesis is the most plausible?
Basic hypotheses

It seems that parameterized complexity theory cannot be built on assuming $P \neq NP$ – we have to assume something stronger.

Let us choose a basic hypothesis:

**Engineers’ Hypothesis**

$k$-CLIQUE cannot be solved in time $f(k) \cdot n^{O(1)}$.

**Theorists’ Hypothesis**

$k$-STEP HALTING PROBLEM (is there a path of the given NTM that stops in $k$ steps?) cannot be solved in time $f(k) \cdot n^{O(1)}$.

**Exponential Time Hypothesis (ETH)**

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

Which hypothesis is the most plausible?
Summary

- **Independent Set** and **k-Step Halting Problem** can be reduced to each other $\Rightarrow$ Engineers’ Hypothesis and Theorists’ Hypothesis are equivalent!

- **Independent Set** and **k-Step Halting Problem** can be reduced to **Dominating Set**.

Is there a parameterized reduction from **Dominating Set** to **Independent Set**? Probably not. Unlike in NP-completeness, where most problems are equivalent, here we have a hierarchy of hard problems.


Does not matter if we only care about whether a problem is FPT or not!
Summary

- **Independent Set** and **k-Step Halting Problem** can be reduced to each other ➞ Engineers’ Hypothesis and Theorists’ Hypothesis are equivalent!
- **Independent Set** and **k-Step Halting Problem** can be reduced to **Dominating Set**.
- Is there a parameterized reduction from **Dominating Set** to **Independent Set**?
  - Probably not. Unlike in **NP**-completeness, where most problems are equivalent, here we have a hierarchy of hard problems.
    - **Independent Set** is **W[1]**-complete.
    - **Dominating Set** is **W[2]**-complete.
- Does not matter if we only care about whether a problem is **FPT** or not!
**Boolean circuit**

A **Boolean circuit** consists of input gates, negation gates, AND gates, OR gates, and a single output gate.

**Circuit Satisfiability**: Given a Boolean circuit $C$, decide if there is an assignment on the inputs of $C$ making the output true.
Boolean circuit

A **Boolean circuit** consists of input gates, negation gates, AND gates, OR gates, and a single output gate.

---

**Circuit Satisfiability:** Given a Boolean circuit $C$, decide if there is an assignment on the inputs of $C$ making the output true.

**Weight of an assignment:** number of true values.

**Weighted Circuit Satisfiability:** Given a Boolean circuit $C$ and an integer $k$, decide if there is an assignment of weight $k$ making the output true.