Fine-Grained Complexity and Algorithm Design Boot Camp

Parameterized Reductions

Dániel Marx

Institute for Computer Science and Control,
Hungarian Academy of Sciences (MTA SZTAKI)
Budapest, Hungary

Simons Institute, Berkeley, CA
September 2, 2015

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?

  $\Rightarrow$ Today

- Can we show that a problem (e.g., VERTEX COVER) has **no** algorithm with running time, say, $2^{o(k)} \cdot n^{O(1)}$?

  $\Rightarrow$ Thursday 3pm

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?

  ⇒ Today
- Can we show that a problem (e.g., Vertex Cover) has **no** algorithm with running time, say, $2^{o(k)} \cdot n^{O(1)}$?

  ⇒ Thursday 3pm

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.

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.

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 ϕ 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 ϕ 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.

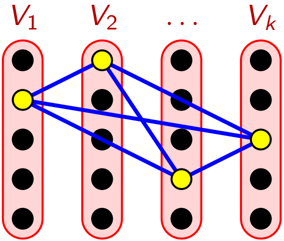
Example: Transforming an INDEPENDENT SET instance (G, k) into a CLIQUE instance $(\overline{G}, k)$ **is** a parameterized reduction.

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)



Theorem

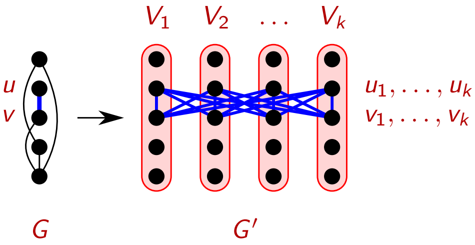
There is a parameterized reduction from Clique to Multicolored Clique.

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.



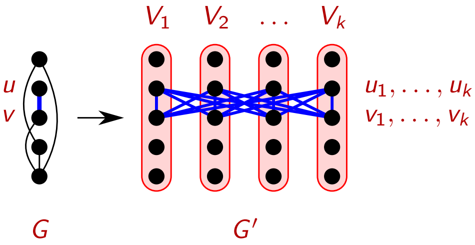
k-clique in $G \iff$ multicolored k-clique in G'.

Multicolored Clique

Theorem

There is a parameterized reduction from 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'.

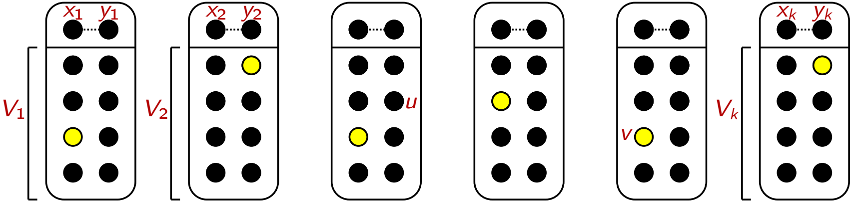
Similarly: reduction to Multicolored Independent Set.

Dominating Set

Theorem

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

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



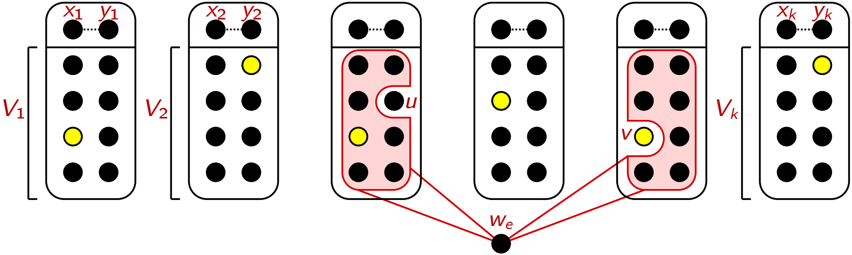
- The dominating set has to contain one vertex from each of the k cliques $V_1, \dots, V_k$ to dominate every x_i and y_i.

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.
- 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.

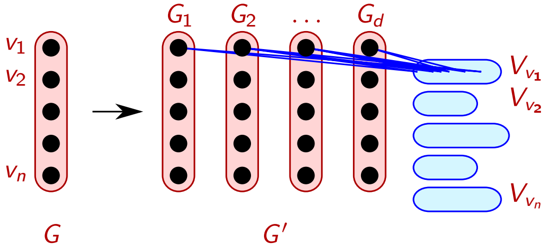
All of these problems are equivalent under parameterized reductions, hence at least as hard as CLIQUE.

Regular graphs

Theorem

There is a parameterized reduction from CLIQUE to CLIQUE on regular graphs.

Proof: Given a graph G and an integer k, let d be the maximum degree of G. Take d copies of G and for every $v \in V(G)$, fully connect every copy of v with a set V_v of $d - d(v)$ vertices.



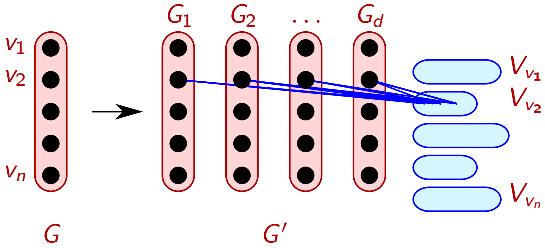
Observe the edges incident to V_v do not appear in any triangle, hence every k-clique of G' is a k-clique of G (assuming $k \geq 3$).

Regular graphs

Theorem

There is a parameterized reduction from CLIQUE to CLIQUE on regular graphs.

Proof: Given a graph G and an integer k, let d be the maximum degree of G. Take d copies of G and for every $v \in V(G)$, fully connect every copy of v with a set V_v of $d - d(v)$ vertices.



Observe the edges incident to V_v do not appear in any triangle, hence every k-clique of G' is a k-clique of G (assuming $k \geq 3$).

Partial Vertex Cover

Partial Vertex Cover: Given a graph G, integers k and s, find k vertices that cover at least s edges.

Theorem

There is a parameterized reduction from Independent Set on regular graphs parameterized by k to Partial Vertex Cover parameterized by k.

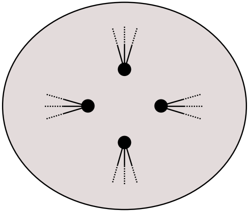
Partial Vertex Cover

Partial Vertex Cover: Given a graph G, integers k and s, find k vertices that cover at least s edges.

Theorem

There is a parameterized reduction from Independent Set on regular graphs parameterized by k to Partial Vertex Cover parameterized by k.

Proof: If G is d-regular, then k vertices can cover $s := kd$ edges if and only if there is a independent set of size k.



$d = 3$, $k = 4$, $s = 12$

Hard problems

Hundreds of parameterized problems are known to be at least as hard as CLIQUE:

- INDEPENDENT SET
- SET COVER
- HITTING SET
- CONNECTED DOMINATING SET
- INDEPENDENT DOMINATING SET
- PARTIAL VERTEX COVER parameterized by k
- DOMINATING SET in bipartite graphs
- ...

We believe that none of these problems are FPT.

Basic hypotheses

It seems that parameterized complexity theory cannot be built on assuming $\mathsf{P} \neq \mathsf{NP}$ – 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 $\mathsf{P} \neq \mathsf{NP}$ – 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 $\mathrm{P} \neq \mathrm{NP}$ – 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?

Independent Set ⇒ Turing machines

Theorem

There is a parameterized reduction from Independent Set to the k-Step Halting Problem.

Proof: Given a graph G and an integer k, we construct a Turing machine M and an integer $k' = O(k^2)$ such that M halts in k' steps if and only if G has an independent set of size k.

INDEPENDENT SET ⇒ Turing machines

Theorem

There is a parameterized reduction from INDEPENDENT SET to the k-STEP HALTING PROBLEM.

Proof: Given a graph G and an integer k, we construct a Turing machine M and an integer $k' = O(k^2)$ such that M halts in k' steps if and only if G has an independent set of size k.

The alphabet Σ of M is the set of vertices of G.

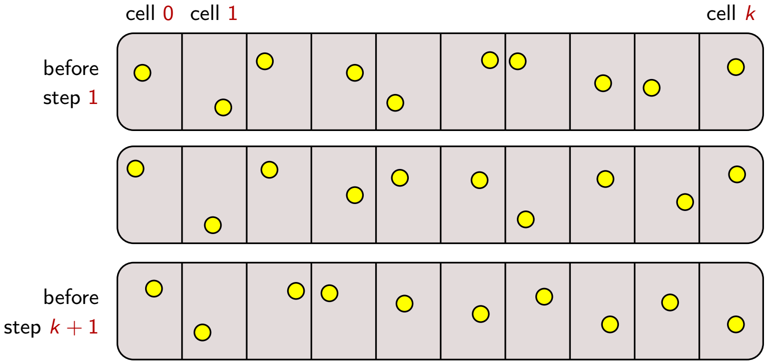
- In the first k steps, M nondeterministically writes k vertices to the first k cells.
- For every $1 \leq i \leq k$, M moves to the i-th cell, stores the vertex in the internal state, and goes through the tape to check that every other vertex is nonadjacent with the i-th vertex (otherwise M loops).
- M does k checks and each check can be done in $2k$ steps $\Rightarrow$ $k' = O(k^2)$.

Turing machines ⇒ Independent Set

Theorem

There is a parameterized reduction from the k-Step Halting Problem to Independent Set.

Proof: Given a Turing machine M and an integer k, we construct a graph G that has an independent set of size $k' := (k+1)^2$ if and only if M halts in k steps.



Turing machines ⇒ Independent Set

Theorem

There is a parameterized reduction from the k-Step Halting Problem to Independent Set.

Proof: Given a Turing machine M and an integer k, we construct a graph G that has an independent set of size $k' := (k+1)^2$ if and only if M halts in k steps.

- G consists of $(k+1)^2$ cliques, thus a k'-independent set has to contain one vertex from each.
- The selected vertex from clique $K_{i,j}$ describes the situation before step i at cell j: what is written there, is the head there, and if so, what the state is, and what the next transition is.
- We add edges between the cliques to rule out inconsistencies: head is at more than one location at the same time, wrong character is written, head moves in the wrong direction etc.

Summary

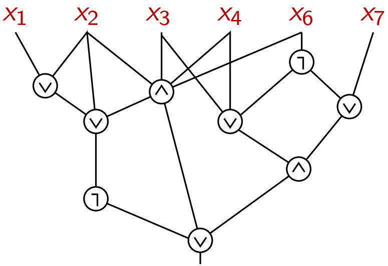
- INDEPENDENT SET and k-STEP HALTING PROBLEM can be reduced to each other $\Rightarrow$ Engineers' Hypothesis and Theorists' Hypothesis are equivalent!
- INDEPENDENT SET and k-STEP HALTING PROBLEM can be reduced to DOMINATING SET.

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

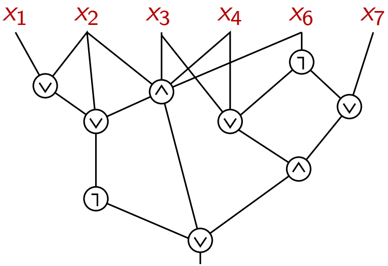
A **Boolean circuit** consists of input gates, negation gates, AND gates, OR gates, and a single output gate.



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.



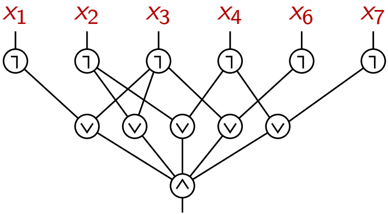
CIRCUIT SATISFIABILITY: Given a Boolean circuit C, decide if there is an assignment on the inputs of C making the output true.

Weight of an assignment: number of true values.

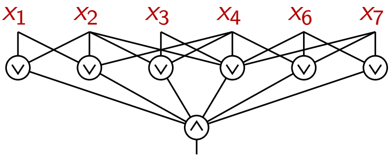
WEIGHTED CIRCUIT SATISFIABILITY: Given a Boolean circuit C and an integer k, decide if there is an assignment of weight k making the output true.

Weighted Circuit Satisfiability

Independent Set can be reduced to Weighted Circuit Satisfiability:

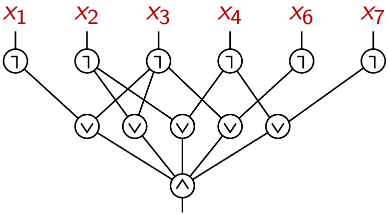


Dominating Set can be reduced to Weighted Circuit Satisfiability:

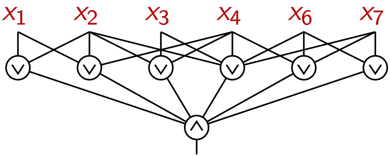


Weighted Circuit Satisfiability

Independent Set can be reduced to Weighted Circuit Satisfiability:



Dominating Set can be reduced to Weighted Circuit Satisfiability:



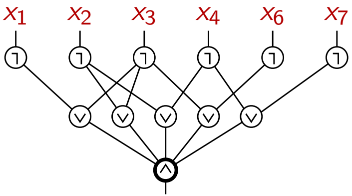
To express Dominating Set, we need more complicated circuits.

Depth and weft

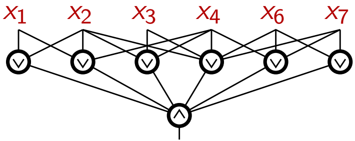
The **depth** of a circuit is the maximum length of a path from an input to the output.

A gate is **large** if it has more than 2 inputs. The **weft** of a circuit is the maximum number of large gates on a path from an input to the output.

Independent Set: weft 1, depth 3



Dominating Set: weft 2, depth 2



The W-hierarchy

Let $C[t, d]$ be the set of all circuits having weft at most t and depth at most d.

Definition

A problem P is in the class W[t] if there is a constant d and a parameterized reduction from P to WEIGHTED CIRCUIT SATISFIABILITY of $C[t, d]$.

We have seen that INDEPENDENT SET is in W[1] and DOMINATING SET is in W[2].

Fact: INDEPENDENT SET is W[1]-complete.
Fact: DOMINATING SET is W[2]-complete.

The W-hierarchy

Let $C[t, d]$ be the set of all circuits having weft at most t and depth at most d.

Definition

A problem P is in the class W[t] if there is a constant d and a parameterized reduction from P to WEIGHTED CIRCUIT SATISFIABILITY of $C[t, d]$.

We have seen that INDEPENDENT SET is in W[1] and DOMINATING SET is in W[2].

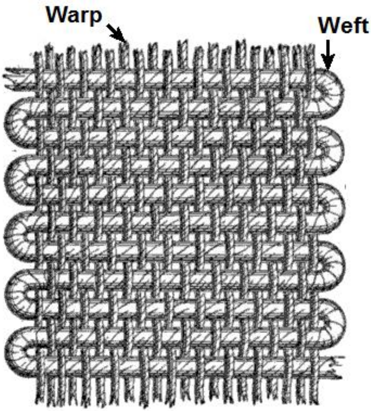
Fact: INDEPENDENT SET is W[1]-complete.
Fact: DOMINATING SET is W[2]-complete.

If any W[1]-complete problem is FPT, then FPT = W[1] and **every** problem in W[1] is FPT.

If any W[2]-complete problem is in W[1], then W[1] = W[2].

$\Rightarrow$ If there is a parameterized reduction from DOMINATING SET to INDEPENDENT SET, then W[1] = W[2].

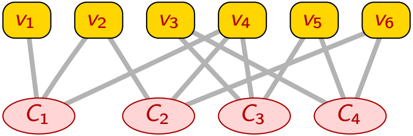
Weft



Weft is a term related to weaving cloth: it is the thread that runs from side to side in the fabric.

Parameterized reductions

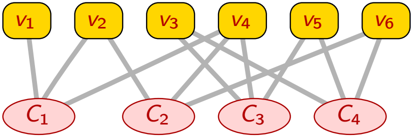
Typical NP-hardness proofs: reduction from e.g., CLIQUE or 3SAT, representing each vertex/edge/variable/clause with a gadget.



Usually does not work for parameterized reductions: cannot afford the parameter increase.

Parameterized reductions

Typical NP-hardness proofs: reduction from e.g., CLIQUE or 3SAT, representing each vertex/edge/variable/clause with a gadget.



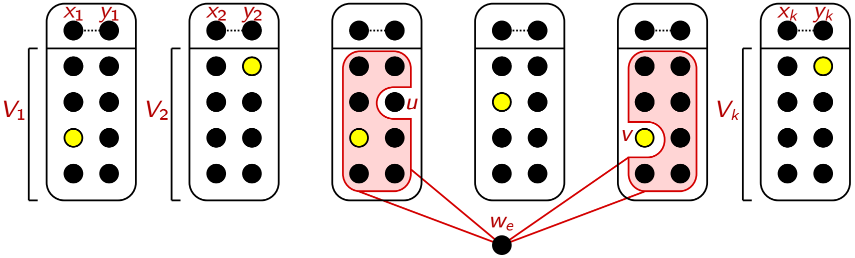
Usually does not work for parameterized reductions: cannot afford the parameter increase.

Types of parameterized reductions:

- Reductions keeping the structure of the graph.
  - CLIQUE ⇒ INDEPENDENT SET
  - INDEPENDENT SET on regular graphs ⇒ PARTIAL VERTEX COVER
- Reductions with vertex representations.
  - MULTICOLORED INDEPENDENT SET ⇒ DOMINATING SET
- Reductions with vertex and edge representations.

Vertex representation

Recall: Reduction from MULTICOLORED INDEPENDENT SET to DOMINATING SET:



List Coloring

List Coloring is a generalization of ordinary vertex coloring: given a

- graph G,
- a set of colors C, and
- a list $L(v) \subseteq C$ for each vertex v,

the task is to find a coloring c where $c(v) \in L(v)$ for every v.

Theorem

Vertex Coloring is FPT parameterized by treewidth.

However, list coloring is more difficult:

Theorem

List Coloring is W[1]-hard parameterized by treewidth.

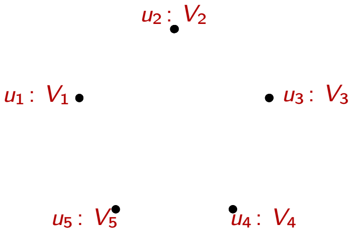
List Coloring

Theorem

List Coloring is W[1]-hard parameterized by treewidth.

Proof: By reduction from Multicolored Independent Set.

- Let G be a graph with color classes $V_1, \ldots, V_k$.
- Set C of colors: the set of vertices of G.
- The colors appearing on vertices $u_1, \ldots, u_k$ correspond to the k vertices of the clique, hence we set $L(u_i) = V_i$.



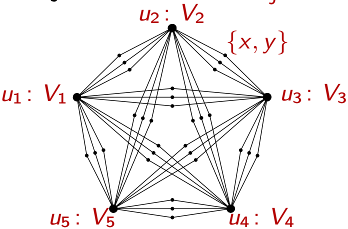
List Coloring

Theorem

List Coloring is W[1]-hard parameterized by treewidth.

Proof: By reduction from Multicolored Independent Set.

- Let G be a graph with color classes $V_1, \ldots, V_k$.
- Set C of colors: the set of vertices of G.
- The colors appearing on vertices $u_1, \ldots, u_k$ correspond to the k vertices of the clique, hence we set $L(u_i) = V_i$.
- If $x \in V_i$ and $y \in V_j$ are adjacent in G, then we need to ensure that $c(u_i) = x$ and $c(u_j) = y$ are not true at the same time $\Rightarrow$ we add a vertex adjacent to u_i and u_j whose list is $\{x, y\}$.



Vertex representation

Key idea

- Represent the k vertices of the solution with k gadgets.
- Connect the gadgets in a way that ensures that the represented values are **compatible.**

But sometimes it is very difficult to create connections that force two gadgets to be compatible. . .

Odd Set

Odd Set: Given a set system $\mathcal{F}$ over a universe U and an integer k, find a set S of at most k elements such that $|S \cap F|$ is odd for every $F \in \mathcal{F}$.

Theorem

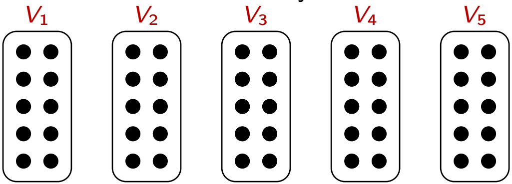
Odd Set is W[1]-hard parameterized by k.

Odd Set

Theorem

Odd Set is W[1]-hard parameterized by k.

First try: Reduction from Multicolored Independent Set.
Let $U = V_1 \cup \ldots V_k$ and introduce each set V_i into $\mathcal{F}$.
$\Rightarrow$ The solution has to contain exactly one element from each V_i.



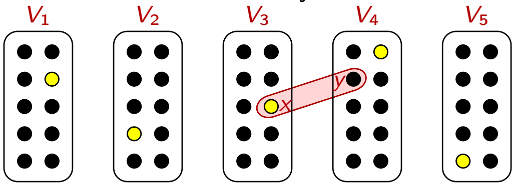
If $xy \in E(G)$, how can we express that $x \in V_i$ and $y \in V_j$ cannot be selected simultaneously?

Odd Set

Theorem

Odd Set is W[1]-hard parameterized by k.

First try: Reduction from Multicolored Independent Set.
Let $U = V_1 \cup \ldots V_k$ and introduce each set V_i into $\mathcal{F}$.
$\Rightarrow$ The solution has to contain exactly one element from each V_i.



If $xy \in E(G)$, how can we express that $x \in V_i$ and $y \in V_j$ cannot be selected simultaneously? Seems difficult:

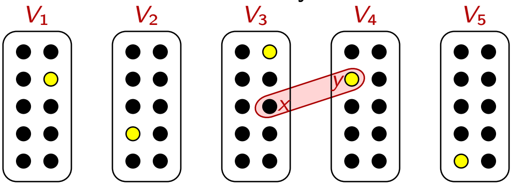
- introducing $\{x, y\}$ into $\mathcal{F}$ forces that **exactly one** of x and y appears in the solution,

Odd Set

Theorem

Odd Set is W[1]-hard parameterized by k.

First try: Reduction from Multicolored Independent Set.
Let $U = V_1 \cup \ldots V_k$ and introduce each set V_i into $\mathcal{F}$.
$\Rightarrow$ The solution has to contain exactly one element from each V_i.



If $xy \in E(G)$, how can we express that $x \in V_i$ and $y \in V_j$ cannot be selected simultaneously? Seems difficult:

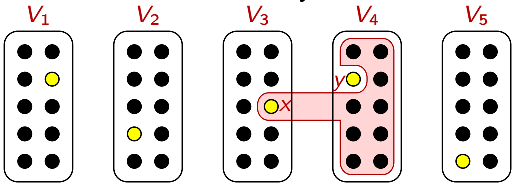
- introducing $\{x, y\}$ into $\mathcal{F}$ forces that **exactly one** of x and y appears in the solution,

Odd Set

Theorem

Odd Set is W[1]-hard parameterized by k.

First try: Reduction from Multicolored Independent Set.
Let $U = V_1 \cup \ldots V_k$ and introduce each set V_i into $\mathcal{F}$.
$\Rightarrow$ The solution has to contain exactly one element from each V_i.



If $xy \in E(G)$, how can we express that $x \in V_i$ and $y \in V_j$ cannot be selected simultaneously? Seems difficult:

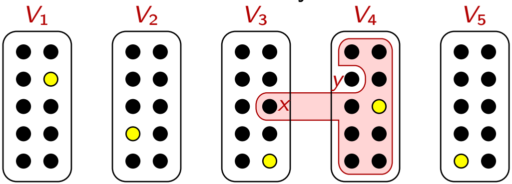
- introducing $\{x, y\}$ into $\mathcal{F}$ forces that **exactly one** of x and y appears in the solution,
- introducing $\{x\} \cup (V_j \setminus \{y\})$ into $\mathcal{F}$ forces that either **both** x and y or **none** of x and y appear in the solution.

Odd Set

Theorem

Odd Set is W[1]-hard parameterized by k.

First try: Reduction from Multicolored Independent Set.
Let $U = V_1 \cup \ldots V_k$ and introduce each set V_i into $\mathcal{F}$.
$\Rightarrow$ The solution has to contain exactly one element from each V_i.



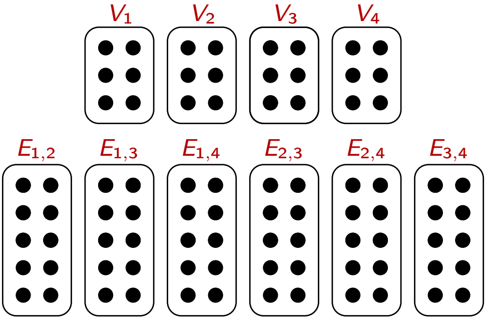
If $xy \in E(G)$, how can we express that $x \in V_i$ and $y \in V_j$ cannot be selected simultaneously? Seems difficult:

- introducing $\{x, y\}$ into $\mathcal{F}$ forces that **exactly one** of x and y appears in the solution,
- introducing $\{x\} \cup (V_j \setminus \{y\})$ into $\mathcal{F}$ forces that either **both** x and y or **none** of x and y appear in the solution.

Odd Set

Reduction from Multicolored Clique.

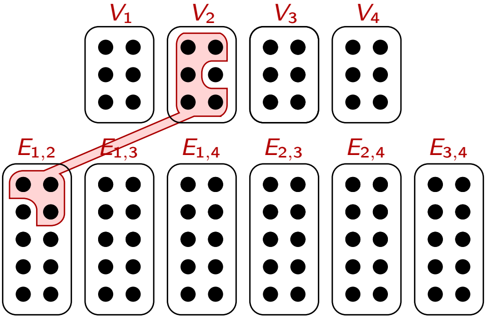
- $U := \bigcup_{i=1}^{k} V_i \cup \bigcup_{1 \le i < j \le k} E_{i,j}$.
- $k' := k + \binom{k}{2}$.
- Let $\mathcal{F}$ contain V_i ($1 \le i \le k$) and $E_{i,j}$ ($1 \le i < j \le k$).



Odd Set

Reduction from Multicolored Clique.

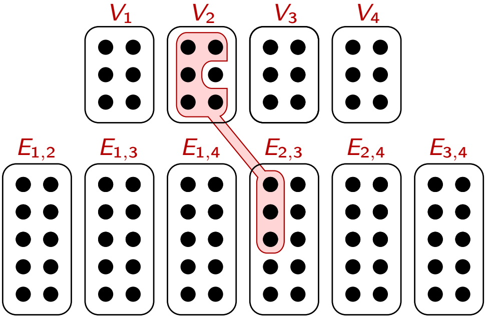
- $U := \bigcup_{i=1}^{k} V_i \cup \bigcup_{1 \le i < j \le k} E_{i,j}$.
- $k' := k + \binom{k}{2}$.
- Let $\mathcal{F}$ contain V_i $(1 \le i \le k)$ and $E_{i,j}$ $(1 \le i < j \le k)$.
- For every $v \in V_i$ and $x \neq i$, we introduce the sets:
  $(V_i \setminus \{v\}) \cup \{\text{every edge from } E_{i,x} \text{ with endpoint } v\}$
  $(V_i \setminus \{v\}) \cup \{\text{every edge from } E_{x,i} \text{ with endpoint } v\}$



Odd Set

Reduction from Multicolored Clique.

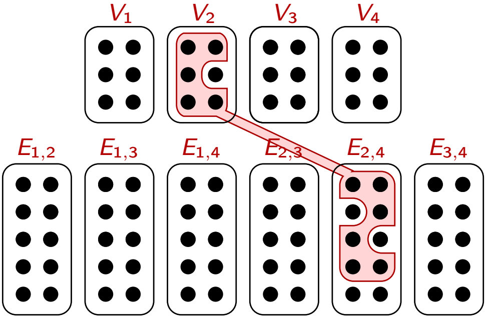
- $U := \bigcup_{i=1}^{k} V_i \cup \bigcup_{1 \le i < j \le k} E_{i,j}$.
- $k' := k + \binom{k}{2}$.
- Let $\mathcal{F}$ contain V_i ($1 \le i \le k$) and $E_{i,j}$ ($1 \le i < j \le k$).
- For every $v \in V_i$ and $x \neq i$, we introduce the sets:
  $(V_i \setminus \{v\}) \cup \{$every edge from $E_{i,x}$ with endpoint $v\}$
  $(V_i \setminus \{v\}) \cup \{$every edge from $E_{x,i}$ with endpoint $v\}$



Odd Set

Reduction from Multicolored Clique.

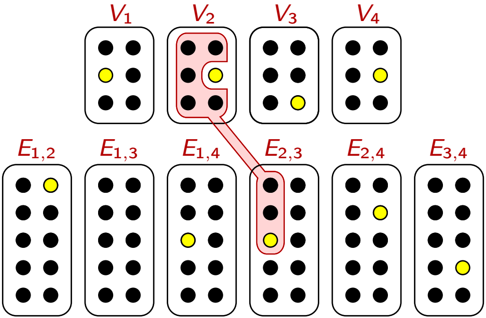
- $U := \bigcup_{i=1}^{k} V_i \cup \bigcup_{1 \le i < j \le k} E_{i,j}$.
- $k' := k + \binom{k}{2}$.
- Let $\mathcal{F}$ contain V_i ($1 \le i \le k$) and $E_{i,j}$ ($1 \le i < j \le k$).
- For every $v \in V_i$ and $x \neq i$, we introduce the sets:
  $(V_i \setminus \{v\}) \cup \{\text{every edge from } E_{i,x} \text{ with endpoint } v\}$
  $(V_i \setminus \{v\}) \cup \{\text{every edge from } E_{x,i} \text{ with endpoint } v\}$



Odd Set

Reduction from Multicolored Clique.

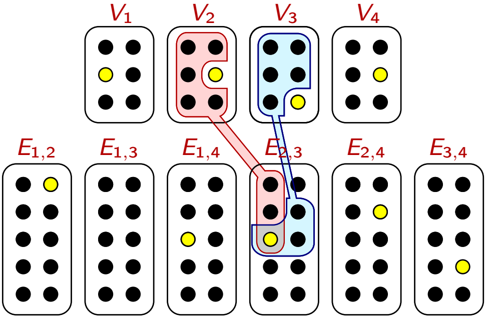
- For every $v \in V_i$ and $x \neq i$, we introduce the sets:
  $(V_i \setminus \{v\}) \cup \{\text{every edge from } E_{i,x} \text{ with endpoint } v\}$
  $(V_i \setminus \{v\}) \cup \{\text{every edge from } E_{x,i} \text{ with endpoint } v\}$
- $v \in V_i$ selected $\iff$ edges with endpoint v are selected from $E_{i,x}$ and $E_{x,i}$



Odd Set

Reduction from Multicolored Clique.

- For every $v \in V_i$ and $x \neq i$, we introduce the sets:
  $(V_i \setminus \{v\}) \cup \{\text{every edge from } E_{i,x} \text{ with endpoint } v\}$
  $(V_i \setminus \{v\}) \cup \{\text{every edge from } E_{x,i} \text{ with endpoint } v\}$
- $v \in V_i$ selected $\iff$ edges with endpoint v are selected from $E_{i,x}$ and $E_{x,i}$
- $v_i \in V_i$ selected, $v_j \in V_j$ selected $\iff$ edge $v_i v_j$ is selected in $E_{i,x}$



Vertex and edge representation

Key idea

- Represent the vertices of the clique by k gadgets.
- Represent the edges of the clique by $\binom{k}{2}$ gadgets.
- Connect edge gadget $E_{i,j}$ to vertex gadgets V_i and V_j such that if $E_{i,j}$ represents the edge between $x \in V_i$ and $y \in V_j$, then it **forces** V_i to x and V_j to y.

The connection between the edge gadget and a vertex gadget needs to express a simple projection relation: a selection of an edge forces a selection of a vertex.

Typically blows up the parameter to $O(k^2)$!

Variants of ODD SET

The following problems are W[1]-hard:

- ODD SET
- EXACT ODD SET (find a set of size exactly k ...)
- EXACT EVEN SET
- UNIQUE HITTING SET
  (at most k elements that hit each set exactly once)
- EXACT UNIQUE HITTING SET
  (exactly k elements that hit each set exactly once)

Variants of ODD SET

The following problems are W[1]-hard:

- ODD SET
- EXACT ODD SET (find a set of size exactly k ...)
- EXACT EVEN SET
- UNIQUE HITTING SET
  (at most k elements that hit each set exactly once)
- EXACT UNIQUE HITTING SET
  (exactly k elements that hit each set exactly once)

Open question:

? EVEN SET: Given a set system $\mathcal{F}$ and an integer k, find a **nonempty** set S of at most k elements such $|F \cap S|$ is even for every $F \in \mathcal{F}$.

Summary

- By parameterized reductions, we can show that lots of parameterized problems are at least as hard as Clique, hence unlikely to be fixed-parameter tractable.
- Connection with Turing machines gives some supporting evidence for hardness (only of theoretical interest).
- The W-hierarchy classifies the problems according to hardness (only of theoretical interest).
- Important trick in W[1]-hardness proofs: vertex and edge representations.