FPT is Characterized by Useful Obstruction Sets

Bart M. P. Jansen

Joint work with Michael R. Fellows, Charles Darwin Univ.
A NEW CHARACTERIZATION OF FPT
Well-Quasi-Orders

- A **quasi-order** is a transitive, reflexive, binary relation \( \leq \) on a (usually infinite) set \( S \).
  - If \( x \leq y \), then \( x \) **precedes** \( y \).
Well-Quasi-Orders

• A **quasi-order** is a transitive, reflexive, binary relation \( \leq \) on a (usually infinite) set \( S \).
  – If \( x \leq y \), then \( x \) precedes \( y \).

• A quasi-order \( \leq \) is a **well-quasi-order** on \( S \) if
  – for every infinite sequence \( x_1, x_2, ... \) over \( S \),
  – there are indices \( i<j \) such that \( x_i \leq x_j \).
Well-Quasi-Orders

• A **quasi-order** is a transitive, reflexive, binary relation \( \leq \) on a (usually infinite) set \( S \).
  – If \( x \leq y \), then \( x \) **precedes** \( y \).

• A quasi-order \( \leq \) is a **well-quasi-order** on \( S \) if
  – for every infinite sequence \( x_1, x_2, \ldots \) over \( S \),
  – there are indices \( i < j \) such that \( x_i \leq x_j \).

• Set \( L \subseteq S \) is a **lower ideal** of \( S \) under \( \leq \) if
  – \( \forall x, y \in S \): if \( x \in L \) and \( y \leq x \), then \( y \in L \).
Well-Quasi-Orders

• A \textbf{quasi-order} is a transitive, reflexive, binary relation \( \leq \) on a (usually infinite) set \( S \).
  – If \( x \leq y \), then \( x \) \textbf{precedes} \( y \).

• A quasi-order \( \leq \) is a \textbf{well-quasi-order} on \( S \) if
  – for every infinite sequence \( x_1, x_2, \ldots \) over \( S \),
  – there are indices \( i<j \) such that \( x_i \leq x_j \).

• Set \( L \subseteq S \) is a \textbf{lower ideal} of \( S \) under \( \leq \) if
  – \( \forall x, y \in S \): if \( x \in L \) and \( y \leq x \), then \( y \in L \).

• A quasi-order is \textbf{polynomial} if \( x \leq y \) can be tested in \( \text{poly}(|x|+|y|) \) time.
The Obstruction Principle

If $\leq$ is a WQO on $S$, and $L \subseteq S$ is a lower ideal, then there is a finite obstruction set $\text{Obs}(L) \subseteq S$, such that for all $x \in S$:

$x \in L$ iff no element in $\text{Obs}(L)$ precedes $x$. 
The Obstruction Principle

If \( \leq \) is a WQO on \( S \), and \( L \subseteq S \) is a lower ideal, then there is a finite obstruction set \( \text{Obs}(L) \subseteq S \), such that for all \( x \in S \):

\[
x \in L \text{ iff no element in } \text{Obs}(L) \text{ precedes } x.
\]

- Decide membership in a lower ideal by testing containment of an obstruction.
The Obstruction Principle

If $\leq$ is a WQO on $S$, and $L \subseteq S$ is a lower ideal, then there is a finite obstruction set $\text{Obs}(L) \subseteq S$, such that for all $x \in S$:

$x \in L$ iff no element in $\text{Obs}(L)$ precedes $x$.

- Decide membership in a lower ideal by testing containment of an obstruction.

- Any element $y \in S \setminus L$ is an obstruction.
The Obstruction Principle

If $\preceq$ is a WQO on $S$, and $L \subseteq S$ is a lower ideal, then there is a finite obstruction set $\text{Obs}(L) \subseteq S$, such that for all $x \in S$: $x \in L$ iff no element in $\text{Obs}(L)$ precedes $x$.

- Decide membership in a lower ideal by testing containment of an obstruction.
- Any element $y \in S \setminus L$ is an obstruction.
- An obstruction is minimal if all elements strictly preceding it belong to $L$. 
Algorithmic Applications of WQO’s

• Fellows & Langston, JACM 1988:
  – $k$-PATH,
  – $k$-VERTEX COVER,
  – $k$-FEEDBACK VERTEX SET,
  can be solved in $O(n^3)$ time, for each fixed $k$. 
Algorithmic Applications of WQO’s

• Fellows & Langston, JACM 1988:
  – $k$-PATH,
  – $k$-VERTEX COVER,
  – $k$-FEEDBACK VERTEX SET,
  can be solved in $O(n^3)$ time, for each fixed $k$.

| Obstruction principle          | Lower ideals                          | Efficient order testing |
|--------------------------------|---------------------------------------|------------------------|
| • Graphs are well-quasi-ordered by the minor relation. | • YES or NO instances are closed under taking minors. | • $f(H)n^3$ time for each fixed graph $H$. |
Algorithmic Applications of WQO’s

• Fellows & Langston, JACM 1988:
  – $k$-PATH,
  – $k$-VERTEX COVER,
  – $k$-FEEDBACK VERTEX SET,
  can be solved in $O(n^3)$ time, for each fixed $k$.

• Results led to the development of parameterized complexity.

| Obstruction principle |
|------------------------|
| • Graphs are well-quasi-ordered by the minor relation. |

| Lower ideals |
|--------------|
| • YES or NO instances are closed under taking minors. |

| Efficient order testing |
|-------------------------|
| • $f(H)n^3$ time for each fixed graph $H$. |
The class FPT
The class FPT

- A parameterized problem is a set $Q \subseteq \Sigma^* \times \mathbb{N}$
  - Each instance $(x,k) \in \Sigma^* \times \mathbb{N}$ has a parameter $k$.
  - The size of an instance $(x,k)$ is $|x| + k$. 

The class FPT

• A **parameterized problem** is a set $Q \subseteq \Sigma^* \times \mathbb{N}$
  – Each instance $(x,k) \in \Sigma^* \times \mathbb{N}$ has a *parameter* $k$.
  – The **size** of an instance $(x,k)$ is $|x| + k$.

**Strongly Uniform FPT (Fixed-Parameter Tractable)**
A parameterized problem $Q$ is strongly uniform FPT if there is an algorithm that decides whether $(x,k) \in Q$ in $f(k)|x|^c$ time.
(for a computable function $f$ and constant $c$)
The class FPT

- A **parameterized problem** is a set $Q \subseteq \Sigma^* \times \mathbb{N}$
  - Each instance $(x,k) \in \Sigma^* \times \mathbb{N}$ has a **parameter** $k$.
  - The **size** of an instance $(x,k)$ is $|x| + k$.

**Strongly Uniform FPT (Fixed-Parameter Tractable)**

A parameterized problem $Q$ is strongly uniform FPT if there is an algorithm that decides whether $(x,k) \in Q$ in $f(k)|x|^c$ time.
(for a computable function $f$ and constant $c$)

- There are weaker notions. (non-uniform, non-computable $f$)
Kernelization

• A **kernel of size** $f(k)$ **for a parameterized problem Q** is a polynomial-time algorithm that transforms $(x,k)$ into $(x',k')$,.
Kernelization

• A **kernel of size** $f(k)$ **for a parameterized problem** $Q$ **is a polynomial-time algorithm** that transforms $(x,k)$ into $(x',k')$. 

\[ (x, k) \quad \xrightarrow{\text{Poly-time}} \quad (x', k') \]
Kernelization

- A **kernel of size** $f(k)$ **for a parameterized problem** $Q$ **is a polynomial-time algorithm** that transforms $(x,k)$ into $(x',k')$, such that $(x,k)$ in $Q$ iff $(x',k')$ in $Q$, Poly-time.
Kernelization

- A **kernel of size** $f(k)$ **for** a parameterized problem $Q$ **is** a polynomial-time algorithm that transforms $(x,k)$ into $(x',k')$,
  - such that $(x,k)$ in $Q$ iff $(x',k')$ in $Q$,
  - and $|x'|+k'$ is bounded by $f(k)$. 

$\leq f(k)$
New characterization of FPT

• For any parameterized problem $Q \subseteq \Sigma^* \times \mathbb{N}$, the following statements are equivalent:
New characterization of FPT

• For any parameterized problem $Q \subseteq \Sigma^* \times \mathbb{N}$, the following statements are equivalent:

1. Problem $Q$ is contained in strongly uniform $\mathbf{FPT}$. 


New characterization of FPT

• For any parameterized problem $Q \subseteq \Sigma^* \times \mathbb{N}$, the following statements are equivalent:

1. Problem $Q$ is contained in strongly uniform $\text{FPT}$.  
2. Problem $Q$ is decidable and admits a \textbf{kernel} of computable size.
For any parameterized problem $Q \subseteq \Sigma^* \times \mathbb{N}$, the following statements are equivalent:

1. Problem $Q$ is contained in strongly uniform $\text{FPT}$.
2. Problem $Q$ is decidable and admits a kernel of computable size.
3. Problem $Q$ is decidable and there is a polynomial-time quasi-order $\preceq$ on $\Sigma^* \times \mathbb{N}$ and a computable function $f: \mathbb{N} \to \mathbb{N}$ such that:
   - The set $Q$ is a lower ideal of $\Sigma^* \times \mathbb{N}$ under $\preceq$.
   - For every $(x,k) \notin Q$, there is an obstruction $(x',k') \notin Q$ of size at most $f(k)$ with $(x',k') \preceq (x,k)$. 
New characterization of FPT

• For any parameterized problem \( Q \subseteq \Sigma^* \times \mathbb{N} \), the following statements are equivalent:

1. Problem \( Q \) is contained in strongly uniform \( \text{FPT} \).
2. Problem \( Q \) is decidable and admits a kernel of computable size.
3. Problem \( Q \) is decidable and there is a polynomial-time quasi-order \( \preceq \) on \( \Sigma^* \times \mathbb{N} \) and a computable function \( f: \mathbb{N} \to \mathbb{N} \) such that:
   • The set \( Q \) is a lower ideal of \( \Sigma^* \times \mathbb{N} \) under \( \preceq \).
   • For every \((x,k) \not\in Q\), there is an obstruction \((x',k') \not\in Q\) of size at most \( f(k) \) with \((x',k') \preceq (x,k)\).
3. Problem $\mathcal{Q}$ is decidable and there is a polynomial-time quasi-order $\leq$ on $\Sigma^* \times \mathbb{N}$ and a computable function $f: \mathbb{N} \rightarrow \mathbb{N}$ such that:

- The set $\mathcal{Q}$ is a lower ideal of $\Sigma^* \times \mathbb{N}$ under $\leq$.
- For every $(x,k) \notin \mathcal{Q}$, there is an obstruction $(x',k') \notin \mathcal{Q}$ of size at most $f(k)$ with $(x',k') \leq (x,k)$. 
New characterization of FPT

• Implies that for every k, there is a **finite** obstruction set $\text{OBS}(k)$ containing instances of size $\leq f(k)$:
  • $(x,k)$ in $Q$ iff no element of $\text{OBS}(k)$ precedes it.

3. Problem $Q$ is decidable and there is a polynomial-time **quasi-order** $\leq$ on $\Sigma^* \times \mathbb{N}$ and a computable function $f: \mathbb{N} \rightarrow \mathbb{N}$ such that:
  • The set $Q$ is a lower ideal of $\Sigma^* \times \mathbb{N}$ under $\leq$.
  • For every $(x,k) \notin Q$, there is an **obstruction** $(x',k') \notin Q$ of size at most $f(k)$ with $(x',k') \leq (x,k)$.
New characterization of FPT

• Implies that for every $k$, there is a finite obstruction set $\text{Obs}(k)$ containing instances of size $\leq f(k)$:
  • $(x,k)$ in $Q$ iff no element of $\text{Obs}(k)$ precedes it.
• The obstruction-testing method that lies at the origins of FPT is not just one way of obtaining FPT algorithms:
  • all of FPT can be obtained this way.

3. Problem $Q$ is decidable and there is a polynomial-time quasi-order $\preceq$ on $\Sigma^* \times \mathbb{N}$ and a computable function $f: \mathbb{N} \rightarrow \mathbb{N}$ such that:
   • The set $Q$ is a lower ideal of $\Sigma^* \times \mathbb{N}$ under $\preceq$.
   • For every $(x,k) \notin Q$, there is an obstruction $(x',k') \notin Q$ of size at most $f(k)$ with $(x',k') \preceq (x,k)$. 
Small kernels yield small obstructions

- Problem $\mathcal{Q}$ is decidable and admits a kernel of size $\mathcal{O}(f(k))$
Small kernels yield small obstructions

- Problem Q is decidable and admits a kernel of size $O(f(k))$ implies
Small kernels yield small obstructions

- Problem $Q$ is decidable and admits a kernel of size $\mathcal{O}(f(k))$

  \textit{implies}

- Problem $Q$ is decidable and there is a polynomial-time quasi-order $\preceq$ on $\Sigma^* \times \mathbb{N}$ such that:
  - The set $Q$ is a lower ideal of $\Sigma^* \times \mathbb{N}$ under $\preceq$.
  - For every $(x,k) \notin Q$, there is an obstruction $(x',k') \notin Q$ of size $\mathcal{O}(f(k))$ with $(x',k') \preceq (x,k)$.
Small kernels yield small obstructions

- Problem $Q$ is decidable and admits a kernel of size $O(f(k))$

implies

- Problem $Q$ is decidable and there is a polynomial-time quasi-order $\leq$ on $\Sigma^* \times \mathbb{N}$ such that:
  - The set $Q$ is a lower ideal of $\Sigma^* \times \mathbb{N}$ under $\leq$.
  - For every $(x,k) \notin Q$, there is an obstruction $(x',k') \notin Q$ of size $O(f(k))$ with $(x',k') \leq (x,k)$.

Parameterized problems with polynomial kernels are characterized by obstructions of polynomial size.
Small kernels yield small obstructions

- Problem Q is decidable and admits a kernel of size $O(f(k))$ implies

- Problem Q is decidable and there is a polynomial-time quasi-order $\preceq$ on $\Sigma^* \times \mathbb{N}$ such that:
  - The set Q is a lower ideal of $\Sigma^* \times \mathbb{N}$ under $\preceq$.
  - For every $(x,k) \notin Q$, there is an obstruction $(x',k') \notin Q$ of size $O(f(k))$ with $(x',k') \preceq (x,k)$.

Parameterized problems with polynomial kernels are characterized by obstructions of polynomial size.

Reverse is false, assuming NP $\not\subset$ coNP/poly. (Kratsch & Walhström, 2011)
Obstruction size vs. kernel size

Polynomial bounds
Obstruction size vs. kernel size

Polynomial bounds

- Best known kernel has $2k - o(k)$ vertices [Lampis’11]
- Largest graph that is minor-minimal with vertex cover size $k$ has $2k$ vertices
- Vertex Cover obstructions have been studied since 1964 [$\alpha$-critical graphs: Erdős, Hajnal & Moon]
Obstruction size vs. kernel size
Polynomial bounds

**$k$-Vertex Cover**
- Best known kernel has $2k - o(k)$ vertices [Lampis’11]
- Largest graph that is minor-minimal with vertex cover size $k$ has $2k$ vertices
  - Vertex Cover obstructions have been studied since 1964 $[\alpha$-critical graphs: Erdős, Hajnal & Moon]

**$k$-$\mathcal{F}$-Minor-Free Deletion (when $\mathcal{F}$ contains a planar graph)**
- Polynomial kernel [Fomin et al.’12]
- Minor-minimal obstructions have polynomial size
Obstruction size vs. kernel size

Polynomial bounds

\( k\text{-}\text{VERTEX COVER} \)

- Best known kernel has \( 2k - o(k) \) vertices [Lampis’11]
- Largest graph that is minor-minimal with vertex cover size \( k \) has \( 2k \) vertices
  - Vertex Cover obstructions have been studied since 1964 [\( \alpha \)-critical graphs: Erdős, Hajnal & Moon]

\( k\text{-}F\text{-MINOR-FREE DELETION} \) (when \( F \) contains a planar graph)

- Polynomial kernel [Fomin et al.’12]
- Minor-minimal obstructions have polynomial size

\( \text{TREewidth parameterized by Vertex Cover} \)

- \( O(vc^3) \)-vertex kernel [Bodlaender et al.’11]
- Minor-minimal obstructions have \( |V| \leq O(vc^3) \).
Obstruction size vs. kernel size

Polynomial bounds

**k-VERTEX COVER**
- Best known kernel has $2k - o(k)$ vertices [Lampis’11]
- Largest graph that is minor-minimal with vertex cover size $k$ has $2k$ vertices
  - Vertex Cover obstructions have been studied since 1964 [α-critical graphs: Erdős, Hajnal & Moon]

**k-\mathcal{F}-MINOR-FREE DELETION (when \mathcal{F} contains a planar graph)**
- Polynomial kernel [Fomin et al.’12]
- Minor-minimal obstructions have polynomial size

**TREewidth parameterized by Vertex Cover**
- $O(vc^3)$-vertex kernel [Bodlaender et al.’11]
- Minor-minimal obstructions have $|V| \leq O(vc^3)$.

**q-COLORING parameterized by Vertex Cover**
- $O(vc^q)$-vertex kernel [J+Kratsch’11]
- Vertex-minimal NO-instances have $vc^{0(q)}$ vertices.
Obstruction size vs. kernel size

Superpolynomial bounds
Obstruction size vs. kernel size
Superpolynomial bounds

3-COLORING parameterized by Feedback Vertex Set

- No polynomial kernel unless $\text{NP} \subseteq \text{coNP/poly}$. [J+Kratsch’11]
- Size of vertex-minimal NO-instances is unbounded in FVS number.
Obstruction size vs. kernel size

Superpolynomial bounds

**3-COLORING** parameterized by Feedback Vertex Set

- No polynomial kernel unless $\text{NP} \subseteq \text{coNP/poly}$. [J+Kratsch’11]
- Size of vertex-minimal NO-instances is unbounded in FVS number.

**$k$-RAMSEY**

- No polynomial kernel unless $\text{NP} \subseteq \text{coNP/poly}$. [Kratsch’12]
- Lower bound construction is based on a Turán-like host graph whose size is superpolynomial in its parameter.
Obstruction size vs. kernel size
Superpolynomial bounds

3-COLORING parameterized by Feedback Vertex Set

• No polynomial kernel unless NP ⊆ coNP/poly. [J+Kratsch’11]
• Size of vertex-minimal NO-instances is unbounded in FVS number.

k-RAMSEY

• No polynomial kernel unless NP ⊆ coNP/poly. [Kratsch’12]
• Lower bound construction is based on a Turán-like host graph whose size is superpolynomial in its parameter.

k-PATHWIDTH

• No polynomial kernel unless NP ⊆ coNP/poly. [BodlaenderDFH’09]
• Minor-minimal obstructions with \( \Omega(3^k) \) vertices.
EXPLOITING OBSTRUCTIONS FOR LOWER-BOUNDS ON KERNEL SIZES
Composition algorithms
Composition algorithms

NP-hard inputs

$X_1$  $X_2$  $X_..$  $X_t$
Composition algorithms

NP-hard inputs

\[ X_1 \rightarrow X_2 \rightarrow \ldots \rightarrow X_n \rightarrow X_{1t} \]
Composition algorithms

NP-hard inputs

\[ \text{poly}(n \cdot t) \text{-time composition} \]
Composition algorithms

NP-hard inputs

\[ \text{poly}(n \cdot t) \text{-time composition} \]

Q-instance

\[ X^* \]

\[ k^* \]
Composition algorithms

NP-hard inputs

\[ \text{poly}(n \cdot t) \]-time composition

Q-instance

\[ k^* \]

poly(n \cdot \log t)
Composition algorithms

NP-hard inputs

poly(n · t)-time composition

Q-instance

AND-Cross-composition: \((x^*, k^*) \in Q \) iff all inputs are YES

OR-Cross-composition: \((x^*, k^*) \in Q \) iff some input is YES
Composition algorithms

NP-hard inputs

poly(n \cdot t)-time composition

Q-instance

poly(n \cdot \log t)
Composition algorithms

- NP-hard inputs
- $X_1$, $X_2$, ..., $X_t$ with poly(n · t)-time composition
- $x^*$ with poly-time poly(k)-size kernel
- $k^*$ with poly(n · log t)
Composition algorithms

NP-hard inputs

\[ x_1 \quad x_2 \quad n \quad x_{..} \quad x_t \]

poly(n \cdot t)-time composition

poly-time poly(k)-size kernel

\[ x^* \quad k^* \quad x' \quad k' \]

poly(n \cdot \log t)
Composition algorithms

NP-hard inputs

\[ x_1 \rightarrow x_2 \rightarrow n \rightarrow x_\ldots \rightarrow x_t \]

poly(n \cdot t)-time composition

poly-time poly(k)-size kernel

\[ x^* \rightarrow k^* \rightarrow x' \rightarrow k' \]

\[ Q\text{-instance} \]

\[ \text{poly}(n \cdot \log t) \]
The \( k \)-Pathwidth problem

- The pathwidth of a graph measures how “path-like” it is
  - Pathwidth does not increase when taking minors
The $k$-Pathwidth problem

- The pathwidth of a graph measures how “path-like” it is
  - Pathwidth does not increase when taking minors

- $k$-PATHWIDTH
  
  **Input:** A graph $G$, an integer $k$.
  
  **Parameter:** $k$.
  
  **Question:** Is the pathwidth of $G$ at most $k$?
The $k$-Pathwidth problem

- The pathwidth of a graph measures how “path-like” it is
  - Pathwidth does not increase when taking minors

- $k$-PATHWIDTH
  **Input:** A graph $G$, an integer $k$.
  **Parameter:** $k$.
  **Question:** Is the pathwidth of $G$ at most $k$?

- Disjoint union acts as AND for question of “pathwidth ≤ $k$?”:
  - $\text{PW}(G_1 \cup G_2 \cup \ldots \cup G_t) \leq k \iff \forall i: \text{PW}(G_i) \leq k$. 
The \textit{k-Pathwidth} problem

- The pathwidth of a graph measures how “path-like” it is
  - Pathwidth does not increase when taking minors

\textit{k-Pathwidth}

\textbf{Input:} A graph \(G\), an integer \(k\).
\textbf{Parameter:} \(k\).
\textbf{Question:} Is the pathwidth of \(G\) at most \(k\)?

- Disjoint union acts as AND for question of “pathwidth \(\leq k\)”:
  - \(\text{PW}(G_1 \cup G_2 \cup ... \cup G_t) \leq k \iff \forall i: \text{PW}(G_i) \leq k\).

- Trivial AND-composition for \textit{k-Pathwidth}:
  - Take disjoint union of \(t\) \textit{Pathwidth}-instances.
    - Ensure same value of \(k\) by padding.
  - Output parameter value is \(k \leq n\).
The $k$-Pathwidth problem

• The pathwidth of a graph measures how “path-like” it is
  – Pathwidth does not increase when taking minors

• $k$-PATHWIDTH
  **Input**: A graph $G$, an integer $k$.
  **Parameter**: $k$.
  **Question**: Is the pathwidth of $G$ at most $k$?

• Disjoint union acts as AND for question of “pathwidth $\leq k$?”:
  – $\text{PW}(G_1 \cup G_2 \cup \ldots \cup G_t) \leq k \iff \forall i: \text{PW}(G_i) \leq k$.

• Trivial AND-composition for $k$-PATHWIDTH:
  – Take disjoint union of $t$ PATHWIDTH-instances.
    • Ensure same value of $k$ by padding.
  – Output parameter value is $k \leq n$.

$k$-PATHWIDTH is AND-compositional and does not admit a polynomial kernel unless $\text{NP} \subseteq \text{coNP/poly}$. [BodlaenderDFH’09,Drucker’12]
OR-Cross-composition
OR-Cross-composition

- The pathwidth measure naturally behaves like an **AND**-gate
**OR-Cross-composition**

- The pathwidth measure naturally behaves like an **AND-gate**
- By exploiting minimal obstructions to $P_{\leq k}$ with $\Omega(3^k)$ vertices, we create an **OR-Cross-composition** of:
  - $t=3^s$ instances of $P_{\text{W-IMPROVEMENT}}(G_1,k), \ldots, (G_t,k)$
  - into one $k$-$P_{\text{ATHWIDTH}}$ instance $(G^*,k^*)$ with $k^* \leq O(n \cdot \log t)$,
  - such that $P_{\text{W}}(G^*) \leq k^*$ iff **some** input $i$ is **YES**.
OR-Cross-composition

• The pathwidth measure naturally behaves like an **AND**-gate

• By exploiting minimal obstructions to \( Pw \leq k \) with \( \Omega(3^k) \) vertices, we create an **OR**-Cross-composition of:
  – \( t=3^s \) instances of \( PW\text{-IMPROVEMENT} \) \((G_1,k), \ldots, (G_t,k)\)
  – into one \( k\text{-PATHWIDTH} \) instance \((G^*,k^*)\) with \( k^* \leq O(n \cdot \log t) \),
  – such that \( PW(G^*) \leq k^* \) iff some input \( i \) is **YES**.

• **PATHWIDTH IMPROVEMENT**

  **Input:** An integer \( k \), and a graph \( G \) of pathwidth \( \leq k-1 \).
  **Question:** Is the pathwidth of \( G \) at most \( k-2 \)?
**OR-Cross-composition**

- The pathwidth measure naturally behaves like an **AND**-gate

- By exploiting minimal obstructions to $Pw\leq k$ with $\Omega(3^k)$ vertices, we create an **OR**-Cross-composition of:
  - $t=3^s$ instances of $Pw$-**Improvement** $(G_1,k), \ldots, (G_t,k)$
  - into one $k$-**Pathwidth** instance $(G^*,k^*)$ with $k^* \leq O(n \cdot \log t)$,
  - such that $Pw(G^*) \leq k^*$ iff some input $i$ is YES.

- **Pathwidth Improvement**
  
  **Input:** An integer $k$, and a graph $G$ of pathwidth $\leq k-1$.
  
  **Question:** Is the pathwidth of $G$ at most $k-2$?

**NP-hard.**
Tree obstructions to Pathwidth

- Kinnersley’92 and TakahashiUK’94 independently proved:
Tree obstructions to Pathwidth

- Kinnersley’92 and TakahashiUK’94 independently proved:
Tree obstructions to Pathwidth

- Kinnersley’92 and TakahashiUK’94 independently proved:

$K_2$ is the unique minimal obstruction to $\text{PW}=0$
Tree obstructions to Pathwidth

- Kinnersley’92 and TakahashiUK’94 independently proved:
Tree obstructions to Pathwidth

- Kinnersley’92 and TakahashiUK’94 independently proved:
Tree obstructions to Pathwidth

- Kinnersley’92 and TakahashiUK’94 independently proved:

Joining 3 minimal tree-obstructions to $\text{PW}=k$, gives minimal obstruction to $\text{PW}=k+1$
Tree obstructions to Pathwidth

• Kinnersley’92 and TakahashiUK’94 independently proved: Joining 3 minimal tree-obstructions to \( PW=k \), gives minimal obstruction to \( PW=k+1 \)
Tree obstructions to Pathwidth

- Kinnersley’92 and TakahashiUK’94 independently proved:
Tree obstructions to Pathwidth

- Kinnersley’92 and TakahashiUK’94 independently proved:
Tree obstructions to Pathwidth

- Kinnersley’92 and TakahashiUK’94 independently proved:
Tree obstructions to Pathwidth

- Kinnersley’92 and TakahashiUK’94 independently proved:

Ternary tree of height $k$, with 1 extra layer of leaves, is minor-minimal obstruction to $PW=k$
Construction
Construction

t=3^s instances of $PW$-IMPROVEMENT with $k=3$
(each asking if $pw(G_i) \leq k - 2$)
Construction

Obstruction with $3^s$ leaves, inflated by factor $k$

Pathwidth is $\mathcal{O}(k \cdot s) \leq \mathcal{O}(n \cdot \log t)$
Construction
Construction
Construction

Output $G^*$ asking for pathwidth $k^*$
1 less than inflated obstruction
Correctness sketch
Correctness sketch

Claim: some input $i$ has $\text{PW}(G_i) \leq k-2 \implies \text{PW}(G^*) < \text{PW}(T^s \diamond k)$
Correctness sketch

Claim: some input $i$ has $PW(G_i) \leq k - 2 \Rightarrow PW(G^*) < PW(T^s \diamond k)$
Correctness sketch
Correctness sketch

Claim: all inputs have $\text{PW}(G_i) > k-2 \implies \text{PW}(G^*) \geq \text{PW}(T^s \diamond k)$
Conclusion
Conclusion

• For each problem in FPT, there is a polynomial-time quasi-order under which each NO-instance \((x,k)\) is preceded by an \(f(k)\)-size obstruction
Conclusion

• For each problem in FPT, there is a polynomial-time quasi-order under which each \texttt{NO}-instance \((x,k)\) is preceded by an \(f(k)\)-size obstruction
• Characterization suggests a connection between kernel sizes and obstruction sizes
Conclusion

• For each problem in FPT, there is a polynomial-time quasi-order under which each NO-instance \((x,k)\) is preceded by an \(f(k)\)-size obstruction
• Characterization suggests a connection between kernel sizes and obstruction sizes
• Large obstructions form the crucial ingredient for OR-cross-composition into \(k\)-PATHWIDTH
Conclusion

• For each problem in FPT, there is a polynomial-time quasi-order under which each NO-instance \((x,k)\) is preceded by an \(f(k)\)-size obstruction

• Characterization suggests a connection between kernel sizes and obstruction sizes

• Large obstructions form the crucial ingredient for OR-cross-composition into \(k\)-PATHWIDTH

• Open problems:
Conclusion

• For each problem in FPT, there is a polynomial-time quasi-order under which each NO-instance (x,k) is preceded by an f(k)-size obstruction
• Characterization suggests a connection between kernel sizes and obstruction sizes
• Large obstructions form the crucial ingredient for OR-cross-composition into $k$-PATHWIDTH

• Open problems:
  1. OR-Cross-composition into $k$-TREEWIDTH?
  2. Further relations between kernel and obstruction sizes?
Conclusion

• For each problem in FPT, there is a polynomial-time quasi-order under which each NO-instance \((x,k)\) is preceded by an \(f(k)\)-size obstruction

• Characterization suggests a connection between kernel sizes and obstruction sizes

• Large obstructions form the crucial ingredient for OR-cross-composition into \(k\)-PATHWIDTH

• Open problems:
  - OR-Cross-composition into \(k\)-TREEWIDTH?
  - Further relations between kernel and obstruction sizes?