Tractable Abstract Argumentation via Backdoor-Treewidth

Wolfgang Dvořák, Markus Hecher, Matthias König,
André Schidler, Stefan Szeider, and Stefan Woltran

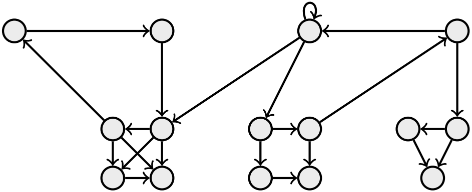
Institute of Logic and Computation, TU Wien, Vienna, Austria

PCCR 2022 (Paper presented at AAAI 2022)

Supported by:
Vienna Science and Technology Fund (WWTF)
Austrian Science Fund (FWF)

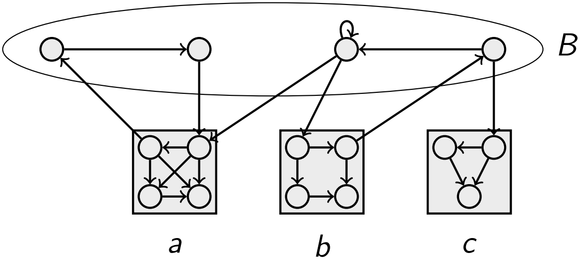
Introduction - Problem

- Argumentation resolves inconsistent information in AI systems
- Standard reasoning tasks computationally expensive
- Exploit structural properties to achieve speedup with **backdoor-treewidth**
- Parameter dominates backdoors and treewidth



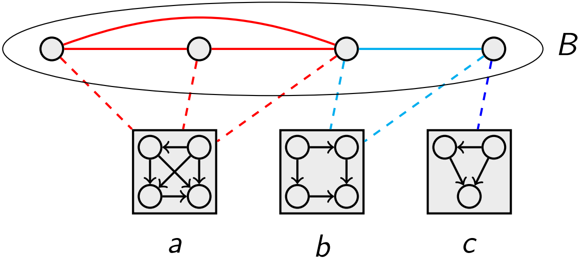
Introduction - Problem

- Argumentation resolves inconsistent information in AI systems
- Standard reasoning tasks computationally expensive
- Exploit structural properties to achieve speedup with **backdoor-treewidth**
- Parameter dominates backdoors and treewidth



Introduction - Problem

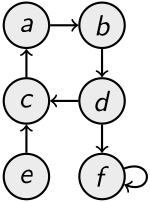
- Argumentation resolves inconsistent information in AI systems
- Standard reasoning tasks computationally expensive
- Exploit structural properties to achieve speedup with **backdoor-treewidth**
- Parameter dominates backdoors and treewidth



Background - AFs

Definition [Dung, AIJ'95]

An AF $F = (A, R)$ consists of **arguments** A and **attacks** $R \subseteq A \times A$.



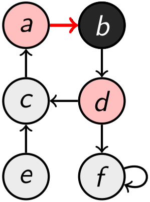
Background - AFs

Definition [Dung, AIJ'95]

An AF $F = (A, R)$ consists of **arguments** A and **attacks** $R \subseteq A \times A$.

A set $S \subseteq A$...

- ...attacks an $a \in A$ if some $b \in S$ s.t. $(b, a) \in R$



$\{a, d\}$ attacks b

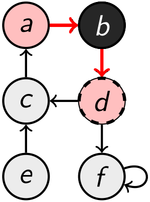
Background - AFs

Definition [Dung, AIJ'95]

An AF $F = (A, R)$ consists of **arguments** A and **attacks** $R \subseteq A \times A$.

A set $S \subseteq A$...

- ...attacks an $a \in A$ if some $b \in S$ s.t. $(b, a) \in R$
- ...defends $a \in A$ if S attacks all attackers of a



$\{a, d\}$ defends d

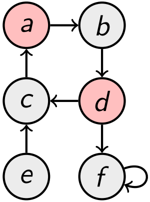
Background - AFs

Definition [Dung, AIJ'95]

An AF $F = (A, R)$ consists of **arguments** A and **attacks** $R \subseteq A \times A$.

A set $S \subseteq A$...

- ...attacks an $a \in A$ if some $b \in S$ s.t. $(b, a) \in R$
- ...defends $a \in A$ if S attacks all attackers of a
- ...is **conflict-free** it attacks no $a \in S$



$\{a, d\}$ is conflict-free

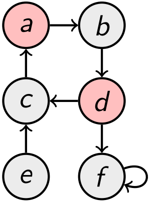
Background - AFs

Definition [Dung, AIJ'95]

An AF $F = (A, R)$ consists of **arguments** A and **attacks** $R \subseteq A \times A$.

A set $S \subseteq A$...

- ...attacks an $a \in A$ if some $b \in S$ s.t. $(b, a) \in R$
- ...defends $a \in A$ if S attacks all attackers of a
- ...is **conflict-free** it attacks no $a \in S$
- ...is **admissible** if c.f. and defends all $a \in S$



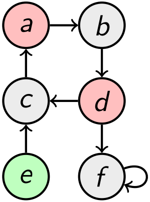
$\{a, d\}$ is admissible

Definition [Dung, AIJ'95]

An AF $F = (A, R)$ consists of **arguments** A and **attacks** $R \subseteq A \times A$.

A set (*extension*) $S \subseteq A$...

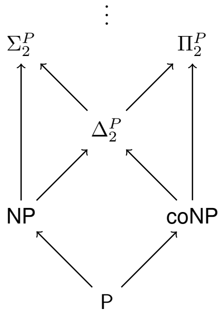
- ...attacks an $a \in A$ if some $b \in S$ s.t. $(b, a) \in R$
- ...defends $a \in A$ if S attacks all attackers of a
- ...is **conflict-free** it attacks no $a \in S$
- ...is **admissible** if c.f. and defends all $a \in S$
- ...is **stable** if c.f. and attacks every $a \in A \setminus E$



$\{a, d\}$ is **not** stable because e is not attacked

Background - Complexity

- We consider credulous/ skeptical reasoning, counting of extensions
- Many problems on first or second level of polynomial hierarchy

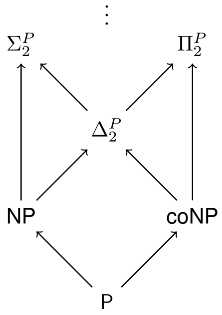


Background - Complexity

- We consider credulous/ skeptical reasoning, counting of extensions
- Many problems on first or second level of polynomial hierarchy
- **Parameterized** algorithms exploit structural properties

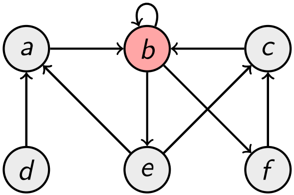
FPT Runtime:

$$O\left(f(p)\cdot n^c\right)$$



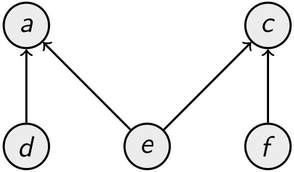
Background - Backdoors & Treewidth

- Backdoor-distance allows reasoning in FPT [Dvorak+Ordyniak+Szeider, AIJ'12]
    - Example: $\{b\}$ is a backdoor to acyclicity



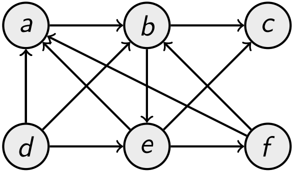
Background - Backdoors & Treewidth

- Backdoor-distance allows reasoning in FPT [Dvorak+Ordyniak+Szeider, AIJ'12]
  - Example: $\{b\}$ is a backdoor to acyclicity



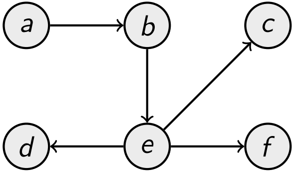
Background - Backdoors & Treewidth

- Backdoor-distance allows reasoning in FPT [Dvorak+Ordyniak+Szeider, AIJ'12]
- Treewidth allows reasoning in FPT [Dvorak+Pichler+Woltran, AIJ'12]
  - Example: high treewidth



Background - Backdoors & Treewidth

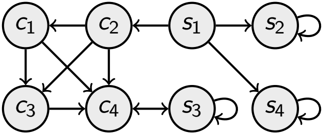
- Backdoor-distance allows reasoning in FPT [Dvorak+Ordyniak+Szeider, AIJ'12]
- Treewidth allows reasoning in FPT [Dvorak+Pichler+Woltran, AIJ'12]
    - Example: small treewidth



Background - Backdoors & Treewidth

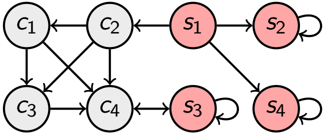
- Backdoor-distance allows reasoning in FPT [Dvorak+Ordyniak+Szeider, AIJ'12]
- Treewidth allows reasoning in FPT [Dvorak+Pichler+Woltran, AIJ'12]
- Backdoor-Treewidth is a combination of both [Ganian+Ramanujan+Szeider, STACS'17]
- Result: Backdoor-Treewidth allows reasoning in FPT with **smaller parameter values**

Backdoor-Treewidth - Example



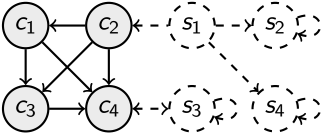
Backdoor-Treewidth - Example

1. Find a backdoor



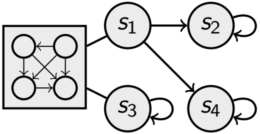
Backdoor-Treewidth - Example

1. Find a backdoor to acyclicity



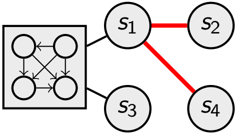
Backdoor-Treewidth - Example

1. Find a backdoor to acyclicity
2. Find the components



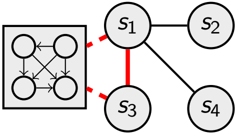
Backdoor-Treewidth - Example

1. Find a backdoor to acyclicity
2. Find the components
3. Construct the torso: **attacks**



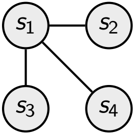
Backdoor-Treewidth - Example

1. Find a backdoor to acyclicity
2. Find the components
3. Construct the torso: **attacks**, **shared components**



Backdoor-Treewidth - Example

1. Find a backdoor to acyclicity
2. Find the components
3. Construct the torso: **attacks**, **shared components**
4. Reason on simple torso graph

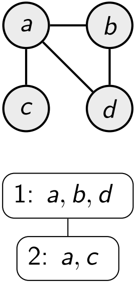


Backdoor-Treewidth

A *tree decomposition (TD)* is a pair $(\mathcal{T}, \mathcal{X})$, where

- $\mathcal{T} = (V_\mathcal{T}, E_\mathcal{T})$ is a tree
- $\mathcal{X} = (X_t)_{t \in V_\mathcal{T}}$ is a set of bags
- $\bigcup_{t \in V_\mathcal{T}} X_t = V$
- for each $v \in V$, the subgraph induced by v in $\mathcal{T}$ is connected
- for each $\{v, w\} \in E$, $\{v, w\} \subseteq X_t$ for some $t \in V_\mathcal{T}$.

Example:

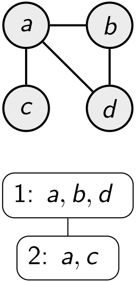


Backdoor-Treewidth

A *tree decomposition (TD)* is a pair $(\mathcal{T}, \mathcal{X})$, where

- $\mathcal{T} = (V_\mathcal{T}, E_\mathcal{T})$ is a tree
- $\mathcal{X} = (X_t)_{t \in V_\mathcal{T}}$ is a set of bags
- $\bigcup_{t \in V_\mathcal{T}} X_t = V$
- for each $v \in V$, the subgraph induced by v in $\mathcal{T}$ is connected
- for each $\{v, w\} \in E$, $\{v, w\} \subseteq X_t$ for some $t \in V_\mathcal{T}$.

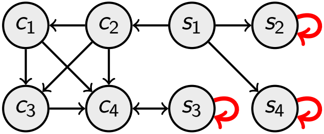
Example:

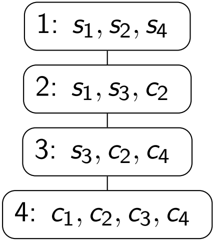


Definition

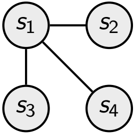
The **backdoor-treewidth** of an AF is the smallest treewidth over **all backdoors** and **all torso-decompositions**.

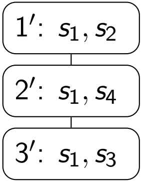
Backdoor-Treewidth - Dominating Parameter





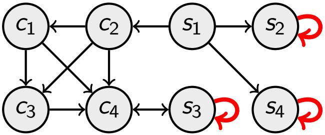
Backdoor: 3
Treewidth: 3





Treewidth: 1

Backdoor-Treewidth - Dominating Parameter



1: s_1, s_2, s_4

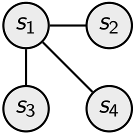
2: s_1, s_3, c_2

3: s_3, c_2, c_4

4: c_1, c_2, c_3, c_4

Backdoor: 3

Treewidth: 3



$1'$: s_1, s_2

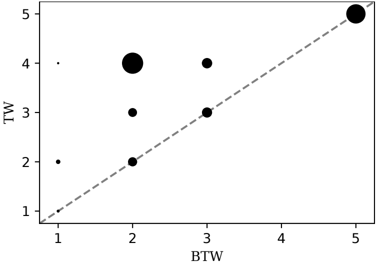
$2'$: s_1, s_4

$3'$: s_1, s_3

Treewidth: 1

Backdoor-Treewidth - In Practice

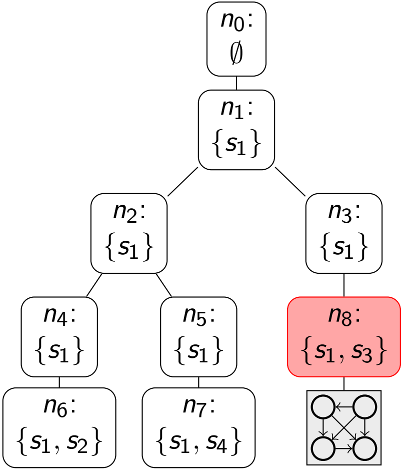
Encoding for exact backdoor-treewidth based on [Samer+Veith, SAT'09].



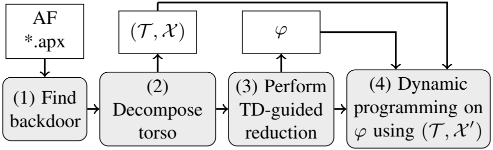
- SAT-formula for each $\texttt{btw}(SF) \leq k$
- $\texttt{btw}(SF) < \texttt{tw}(SF)$ for 611 of the 1134 instances
- $\texttt{btw}(SF) \leq \texttt{tw}(SF)/2$ for 519 out of 611 instances

Exploiting Small Backdoor-Treewidth

- Dynamic Programming on the torso-TD
- Advanced computation in nodes with attached components

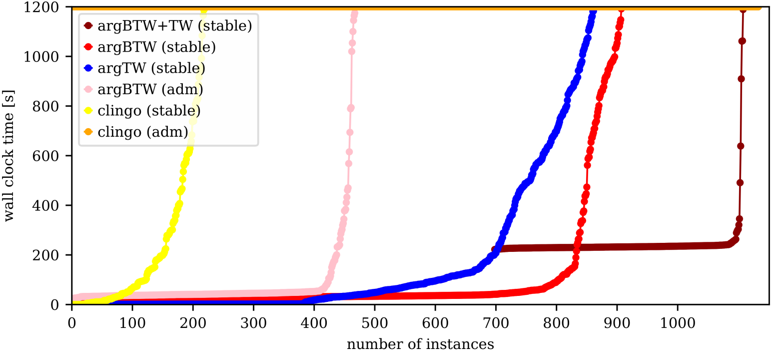


The System argBTW



- `argBTW` implements backdoor-treewidth approach utilizing
  - Heuristics for backdoor and tree-decompositions
  - Tree-decomposition guided reduction to (#)SAT
  - Dynamic Programming on (torso-)decomposition
- Special cases treewidth-only (`argTW`), backdoor-only (`argBW`)
- Supports admissible sets, stable extensions

Performance



Are there instances where `argBTW` outperforms the state-of-the-art?

Summary & Conclusion

- We defined and investigated backdoor-treewidth for abstract argumentation
- We established parameterized tractability by backdoor-treewidth
  - for the fragments acyclicity and even-cycle-freeness
  - for complete and stable extensions
- System `argBTW` outperforms state-of-the-art in certain instances
- Future work: further fragments and semantics, structured approaches