Local Backbones

Ronald de Haan[1], Iyad Kanj[2], Stefan Szeider[1]

[1] Technische Universität Wien
[2] DePaul University

SAT 2013

Backbones in propositional theories

A backbone of a propositional theory is a variable that has the same truth value in each satisfying assignment.

- i.e., $x \in \mathrm{Var}(\varphi)$ is a backbone of a CNF formula φ if $\varphi \models x$ or $\varphi \models \neg x$.

Identifying backbones allows us to simplify the theory.

Unfortunately, deciding whether a variable is a backbone is coNP-complete.

Our approach:

- Relax and localize the notion of a backbone.
- It is reasonable that some variables are enforced locally (local backbones).
- Main theoretical tool: parameterized complexity theory.

Overview

What are local backbones?

Do local backbones occur?

Parameterized complexity results

Iterative local backbones

What are local backbones?

Definition (k-backbones).

A *k-backbone* of a CNF formula φ is a variable $x \in \mathrm{Var}(\varphi)$ such that for some $\varphi' \subseteq \varphi$ with $|\varphi'| \leq k$ it holds that $\varphi' \models x$ or $\varphi' \models \neg x$.

Example: x_2 is a 2-backbone of φ
($\neg x_2$ is implied by a subset of size 2).

$$\varphi = \{\{x_1, \neg x_2\}, \{\neg x_1, \neg x_2\}, \{x_2, x_3, x_4\}, \{x_2, \neg x_3, x_4\}, \{\neg x_4, x_5\}\}$$

- Every k-backbone of φ is a backbone of φ.
- 1-backbones correspond to unit clauses.

Where were we?

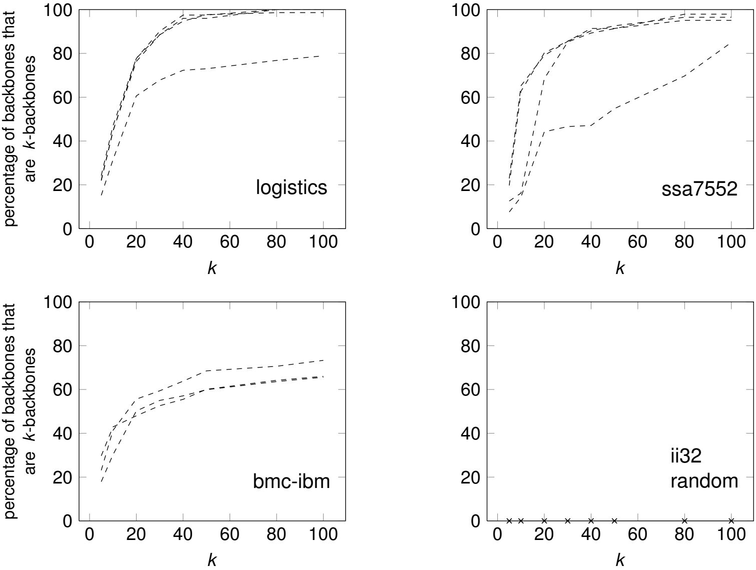
What are local backbones?

Do local backbones occur?

Parameterized complexity results

Iterative local backbones

Distribution of local backbones



Parameterized complexity theory

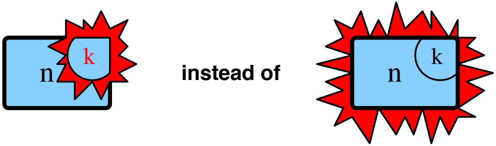
Parameterized complexity theory investigates how to algorithmically exploit structure in problem instances.

- Takes into account a parameter k of the input, besides the input size n.

If k is a constant, then finding k-backbones can be done in polynomial time.

- Brute force search in roughly n^k time (XP).
- For $k = 3, 4, \ldots$ this is already not so practical.

We would like to solve the problem in $f(k) \cdot n^c$ time, for some function f and some constant c: fixed-parameter tractability (FPT).



Parameterized complexity theory

To give evidence that some problems are not FPT,
there exist fixed-parameter intractability classes:

$$\text{FPT} \subseteq \text{W[1]} \subseteq \text{W[2]} \subseteq \cdots \subseteq \text{W[P]}$$

The classes W[t] are based on the question whether certain Boolean circuits are satisfiable with k input nodes set to true.

These classes are not fixed-parameter tractable unless the Exponential Time Hypothesis (ETH) fails.

- ETH: 3SAT cannot be solved in subexponential time.

Where were we?

What are local backbones?

Do local backbones occur?

Parameterized complexity results

Iterative local backbones

The parameterized decision problem

We consider the following parameterized decision problems, for propositional languages $\mathcal{C}$.

LOCAL-BACKBONE[$\mathcal{C}$]

Instance:	*a CNF formula $\varphi \in \mathcal{C}$, a variable $x \in Var(\varphi)$, and an integer $k \geq 1$.*
Parameter:	*k.*
Question:	*Is x a k-backbone of φ?*

Formulas with bounded variable occurrence

Fix an integer $d \geq 1$. We let VO_d denote the class of CNF formulas in which each variable occurs at most d times.

Theorem. LOCAL-BACKBONE(VO_d) is FPT.

Proof (idea). Bounded search tree.

Search for a subset $\varphi' \subseteq \varphi$ witnessing $\varphi' \models \ell$ for some $\ell \in \{x, \neg x\}$ with a bounded search tree.

Start with some clause c containing x.

For each variable y in the current set φ', guess a (non-empty) subset of clauses containing y.

- bounded number of branches, since y occurs in at most d clauses

The depth of the search tree is at most k, since $|\varphi'| \leq k$.

Formulas with bounded variable occurrence

Fix an integer $d \geq 1$. We let VO_d denote the class of CNF formulas in which each variable occurs at most d times.

Theorem. $\textsc{Local-Backbone}(\text{VO}_d)$ is FPT.

Proof (idea). Bounded search tree.

Search for a subset $\varphi' \subseteq \varphi$ witnessing $\varphi' \models \ell$ for some $\ell \in \{x, \neg x\}$ with a bounded search tree.

Start with some clause c containing x.

For each variable y in the current set φ', guess a (non-empty) subset of clauses containing y.

- bounded number of branches,
  since y occurs in at most d clauses

The depth of the search tree is at most k, since $|\varphi'| \leq k$.

Formulas with bounded variable occurrence

Fix an integer $d \geq 1$. We let VO_d denote the class of CNF formulas in which each variable occurs at most d times.

Theorem. Local-Backbone(VO_d) is FPT.

Proof (idea). Bounded search tree.

Example:

$$\varphi = \{\{\neg x_1, x_2\}, \{x_2, x_3\}, \{\neg x_2\}, \{\neg x_3, x_4\}, \{\neg x_3, \neg x_4\}, \{x_4, x_5\}\}$$

$$\{\{x_2, x_3\}\}$$

Formulas with bounded variable occurrence

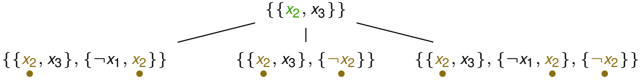
Fix an integer $d \geq 1$. We let VO_d denote the class of CNF formulas in which each variable occurs at most d times.

Theorem. LOCAL-BACKBONE(VO_d) is FPT.

Proof (idea). Bounded search tree.

Example:

$$\varphi = \{\{\neg x_1, x_2\}, \{x_2, x_3\}, \{\neg x_2\}, \{\neg x_3, x_4\}, \{\neg x_3, \neg x_4\}, \{x_4, x_5\}\}$$



Formulas with bounded variable occurrence

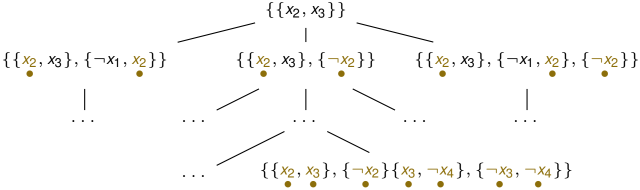
Fix an integer $d \geq 1$. We let VO_d denote the class of CNF formulas in which each variable occurs at most d times.

Theorem. LOCAL-BACKBONE(VO_d) is FPT.

Proof (idea). Bounded search tree.

Example:

$$\varphi = \{\{\neg x_1, x_2\}, \{x_2, x_3\}, \{\neg x_2\}, \{\neg x_3, x_4\}, \{\neg x_3, \neg x_4\}, \{x_4, x_5\}\}$$



Formulas with bounded variable occurrence

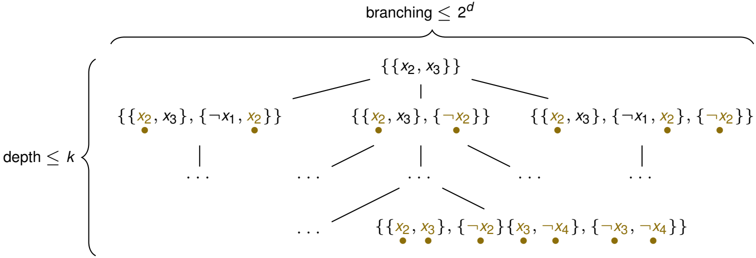
Fix an integer $d \geq 1$. We let VO_d denote the class of CNF formulas in which each variable occurs at most d times.

Theorem. LOCAL-BACKBONE(VO_d) is FPT.

Proof (idea). Bounded search tree.

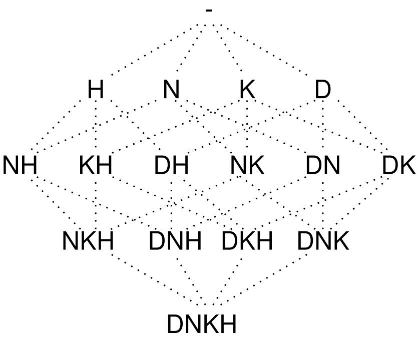
Example:

$$\varphi = \{\{\neg x_1, x_2\}, \{x_2, x_3\}, \{\neg x_2\}, \{\neg x_3, x_4\}, \{\neg x_3, \neg x_4\}, \{x_4, x_5\}\}$$



Local backbones of various propositional fragments

Complexity of LOCAL-BACKBONE[$\mathcal{C}$], for $\mathcal{C} \subseteq \{D,N,K,H\}$:



D: no purely negative clauses

N: no unit clauses

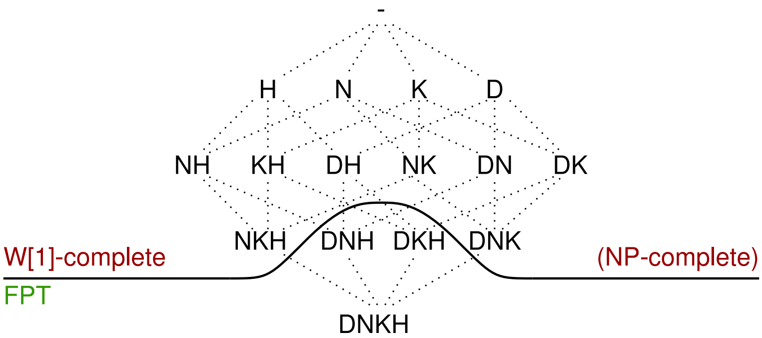
K: clauses are Krom

H: clauses are Horn

e.g., DH corresponds to definite Horn

Local backbones of various propositional fragments

Complexity of $\textsc{Local-Backbone}[\mathcal{C}]$, for $\mathcal{C} \subseteq \{$D,N,K,H$\}$:



D: no purely negative clauses

N: no unit clauses

K: clauses are Krom

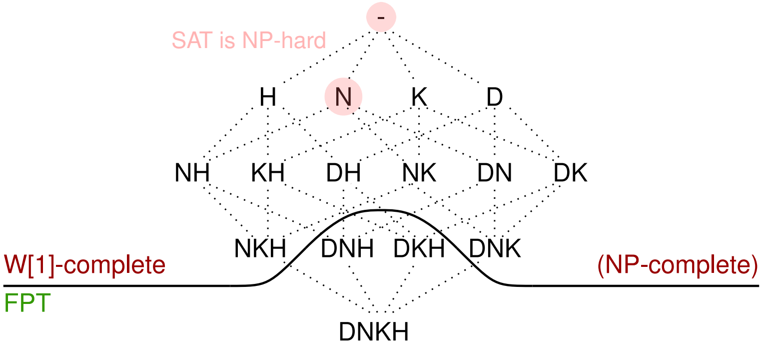
H: clauses are Horn

e.g., DH corresponds to definite Horn

(All results hold also for the restriction to 3CNF.)

Local backbones of various propositional fragments

Complexity of LOCAL-BACKBONE[$\mathcal{C}$], for $\mathcal{C} \subseteq \{$D,N,K,H$\}$:



D: no purely negative clauses

N: no unit clauses

K: clauses are Krom

H: clauses are Horn

e.g., DH corresponds to definite Horn

(All results hold also for the restriction to 3CNF.)

Small Unsatisfiable Subsets

Local backbones are closely related to small unsatisfiable subsets.

- useful for the repair of inconsistent knowledge bases.

Originally considered in Fellows et al. (2006).

SMALL-UNSATISFIABLE-SUBSET[$\mathcal{C}$]

Instance:	*a CNF formula $\varphi \in \mathcal{C}$, and an integer $k \geq 1$.*
Parameter:	*k.*
Question:	*Is there an unsatisfiable $\varphi' \subseteq \varphi$ with at most k clauses?*

Theorem.

For any $\mathcal{C}$, SMALL-UNSATISFIABLE-SUBSET[$\mathcal{C}$] has the same parameterized complexity as LOCAL-BACKBONE[$\mathcal{C}$].

Small Unsatisfiable Subsets

Local backbones are closely related to small unsatisfiable subsets.

- useful for the repair of inconsistent knowledge bases.

Originally considered in Fellows et al. (2006).

SMALL-UNSATISFIABLE-SUBSET[$\mathcal{C}$]

Instance: a CNF formula $\varphi \in \mathcal{C}$, and an integer $k \geq 1$.

Parameter: k.

Question: Is there an unsatisfiable $\varphi' \subseteq \varphi$ with at most k clauses?

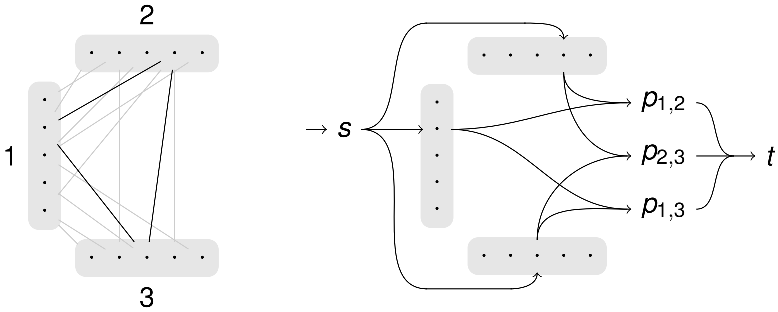
Theorem.

For any $\mathcal{C}$, SMALL-UNSATISFIABLE-SUBSET[$\mathcal{C}$] has the same parameterized complexity as LOCAL-BACKBONE[$\mathcal{C}$].

Definite Horn Formulas

Theorem. LOCAL-BACKBONE[DefHorn] is W[1]-hard.

Proof (idea). Reduction from MULTICOLORED-CLIQUE (see below).



(A slight modification of the proof works for the case of NH.)

Krom Formulas

Remember: deciding whether a variable is a (non-local) backbone of a Krom formula φ can be done in poly-time.

- e.g., by using reachability in the implication graph of φ.

Theorem. LOCAL-BACKBONE[Krom] is W[1]-hard.

Proof (idea). Reduction from CLIQUE.

Essential to the proof:

- paths in the implication graph may use some clauses twice,
- so k-reachability in the implication graph cannot be used.

This contrasts to the result of Buresh-Oppenheim & Mitchell (2006,2007) that finding a minimum (tree-like) resolution refutation of a Krom formula can be found in poly-time.

- The smallest refutation does not necessarily use the smallest number of clauses.

Krom Formulas

Remember: deciding whether a variable is a (non-local) backbone of a Krom formula φ can be done in poly-time.

- e.g., by using reachability in the implication graph of φ.

Theorem. LOCAL-BACKBONE[Krom] is W[1]-hard.

Proof (idea). Reduction from CLIQUE.

Essential to the proof:

- paths in the implication graph may use some clauses twice,
- so k-reachability in the implication graph cannot be used.

This contrasts to the result of Buresh-Oppenheim & Mitchell (2006,2007) that finding a minimum (tree-like) resolution refutation of a Krom formula can be found in poly-time.

- The smallest refutation does not necessarily use the smallest number of clauses.

Where were we?

What are local backbones?

Do local backbones occur?

Parameterized complexity results

Iterative local backbones

Iterative Local Backbones

Iterative local backbones are variables with an enforced truth value that can be found after iteratively instantiating local backbones.

Definition. Iterative k-backbones.

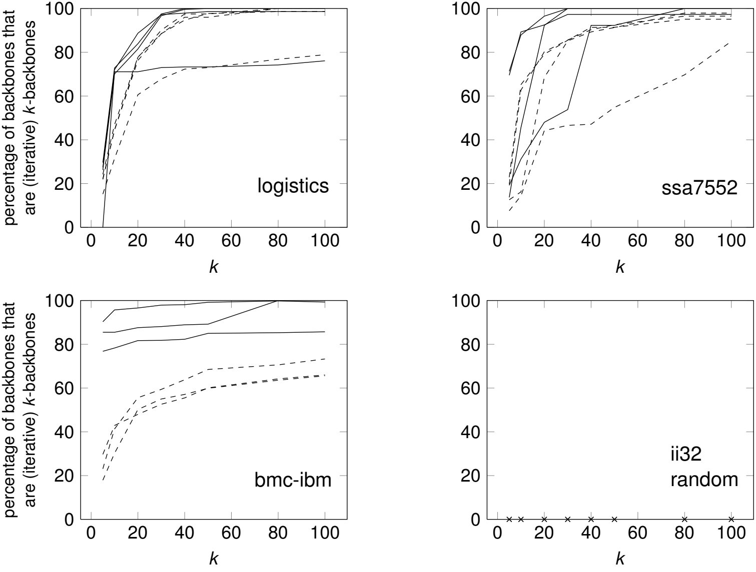
An *iterative k-backbone* of a CNF formula φ is a variable $x \in \mathrm{Var}(\varphi)$ such that either:

- x is a k-backbone of φ; or
- there exists a k-backbone y of φ, with enforced literal $\ell \in \{y, \neg y\}$, and x is an iterative k-backbone of $\varphi|_\ell$.

Example: x_4 is an iterative 2-backbone of φ
($\neg x_2$ is implied by a subset of φ of size 2;
x_4 is implied by a subset of $\varphi|_{\neg x_2}$ of size 2).

$$\varphi = \{\{x_1, \neg x_2\}, \{\neg x_1, \neg x_2\}, \{\cancel{x_2}, x_3, x_4\}, \{\cancel{x_2}, \neg x_3, x_4\}, \{\neg x_4, x_5\}\}$$

Distribution of (iterative) local backbones



Dashed: local backbones, solid: iterative local backbones.

Iterative Local Backbones

For propositional languages $\mathcal{C}$, the parameterized decision problems ITERATIVE-LOCAL-BACKBONE[$\mathcal{C}$] are defined analogously to LOCAL-BACKBONE[$\mathcal{C}$].

Theorem. For any $\mathcal{C}$, if LOCAL-BACKBONE[$\mathcal{C}$] is FPT, then also ITERATIVE-LOCAL-BACKBONE[$\mathcal{C}$] is FPT.

Proof (idea). Iteratively find k-backbones and instantiate them, until a fixed-point is reached.

Theorem. ITERATIVE-LOCAL-BACKBONE[NH] is W[1]-hard.

Proof (idea). The hardness proof for LOCAL-BACKBONE[NH] also works for this case.

Iterative Local Backbones

ITERATIVE-LOCAL-BACKBONE[Krom] is in P.

Proof (idea). Iterative *k*-backbones of a Krom formula φ can be found by iteratively applying backbones that are based on *k*-reachability in the implication graph of φ.

ITERATIVE-LOCAL-BACKBONE[DefHorn] is in P.

Proof (idea). The set of iterative *k*-backbones of a definite Horn formula φ coincides with the set of (non-local) backbones of φ.

!! Remember, LOCAL-BACKBONE[Krom] and LOCAL-BACKBONE[DefHorn] are W[1]-hard

Take home message – quick overview

- Relatively many backbones might be local backbones (or iterative local backbones).
- Identifying local backbones is in XP (poly-time for fixed k).
- For formulas with bounded variable occurrences, it is fixed-parameter tractable.
- It is W[1]-hard already for definite Horn and Krom formulas;
  - interestingly, in these cases iterative local backbones are easier to find (poly-time).
- Finding small unsatisfiable subsets is of the same parameterized complexity (for all fragments).