Strong Backdoors to Nested Satisfiability

Serge Gaspers[1] Stefan Szeider[2]

[1]The University of New South Wales, Sydney, Australia

[2]Vienna University of Technology, Vienna, Austria

SAT 2012

Outline

Result

Theorem (Main result)

SAT *and* #SAT *are fixed-parameter tractable parameterized by the size of a smallest strong backdoor set with respect to the base class of nested CNF formulas.*

Outline

SAT and #SAT

SAT	
Input:	A propositional formula F in conjunctive normal form (CNF)
Question:	Is there an assignment to $\text{var}(F)$ satisfying all clauses of F?

#SAT	
Input:	A CNF formula F
Question:	What is the number of assignment to $\text{var}(F)$ satisfying all clauses of F?

Example:

$$(x_1 \vee x_2) \wedge (\neg x_2 \vee x_3 \vee \neg x_4) \wedge (x_1 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4)$$

SAT: theory vs. practice

theory

- NP-complete
- *ETH*: SAT cannot be solved in time $2^{o(n)}$
- *Strong ETH*: SAT cannot be solved in time $(2-\epsilon)^n$ for any $\epsilon > 0$

practice

- Want to solve an NP-complete problem? Just encode into SAT and use a SAT solver
- *Real-world* instances with millions of variables and clauses

Backdoors

- Belief: real world instances have a "hidden structure" that makes them easy to solve
- Challenge: measure this hidden structure
- One way: *Backdoor* = set of "key" variables that make it easy to solve the formula

Backdoors

- CNF formula F
- Set of variables $B \subseteq \mathrm{var}(F)$
- Base class $\mathcal{C}$: class of poly-time solvable CNF formulas

Definition (Weak Backdoor [Williams, Gomes, Selman, 2003])

B is a weak $\mathcal{C}$-backdoor for F if there is a partial truth assignment τ to B such that $F[\tau] \in \mathcal{C}$ and $F[\tau]$ is satisfiable.

(in $F[\tau]$ all clauses satisfied by τ are removed and the literals on B are removed from all remaining clauses)

Definition (Strong Backdoor [Williams, Gomes, Selman, 2003])

B is a strong $\mathcal{C}$-backdoor for F if for every partial truth assignment τ to B we have $F[\tau] \in \mathcal{C}$.

Backdoors

- CNF formula F
- Set of variables $B \subseteq \text{var}(F)$
- Base class $\mathcal{C}$: class of poly-time solvable CNF formulas

Definition (Weak Backdoor [Williams, Gomes, Selman, 2003])

B is a weak $\mathcal{C}$-backdoor for F if there is a partial truth assignment τ to B such that $F[\tau] \in \mathcal{C}$ and $F[\tau]$ is satisfiable.

(in $F[\tau]$ all clauses satisfied by τ are removed and the literals on B are removed from all remaining clauses)

Definition (Strong Backdoor [Williams, Gomes, Selman, 2003])

B is a strong $\mathcal{C}$-backdoor for F if for every partial truth assignment τ to B we have $F[\tau] \in \mathcal{C}$.

Backdoors

- CNF formula F
- Set of variables $B \subseteq \text{var}(F)$
- Base class $\mathcal{C}$: class of poly-time solvable CNF formulas

Definition (Weak Backdoor [Williams, Gomes, Selman, 2003])

B is a weak $\mathcal{C}$-backdoor for F if there is a partial truth assignment τ to B such that $F[\tau] \in \mathcal{C}$ and $F[\tau]$ is satisfiable.

(in $F[\tau]$ all clauses satisfied by τ are removed and the literals on B are removed from all remaining clauses)

Definition (Strong Backdoor [Williams, Gomes, Selman, 2003])

B is a strong $\mathcal{C}$-backdoor for F if for every partial truth assignment τ to B we have $F[\tau] \in \mathcal{C}$.

Backdoor Problems

Weak/Strong $\mathcal{C}$-Backdoor Detection

Input:	A CNF formula F, an integer k
Question:	Does F have a weak/strong $\mathcal{C}$-backdoor of size at most k?

Weak/Strong $\mathcal{C}$-Backdoor Evaluation

Input:	A CNF formula F, a weak/strong $\mathcal{C}$-backdoor B
Question:	Is F satisfiable?

Outline

Parameterized Complexity

Definition (Parameterized problem)

A parameterized (decision) problem is a subset of $\Sigma^* \times \mathbb{N}$ for some finite alphabet Σ. For an instance $(x,k) \in \Sigma^* \times \mathbb{N}$, x is the main part and k the parameter.

FPT: class of param. pbs that can be solved in time $f(k) \cdot n^{O(1)}$
W[$\cdot$]: parameterized intractability classes
XP: class of param. pbs that can be solved in time $f(k) \cdot n^{g(k)}$

$$\text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \subseteq \ldots \text{XP}.$$

All inclusions believed to be strict.
The classes are closed under parameterized reductions (the parameter of the target problem is upper bounded by a function of the initial parameter).

Parameterized Backdoor Problems

Weak/Strong $\mathcal{C}$-Backdoor Detection

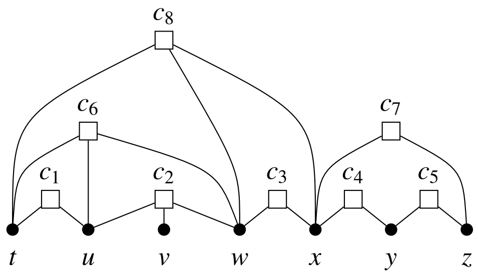
Input:	A CNF formula F, an integer k
Parameter:	k
Question:	Does F have a weak/strong $\mathcal{C}$-backdoor of size at most k?

Weak/Strong $\mathcal{C}$-Backdoor Evaluation

Input:	A CNF formula F, a weak/strong $\mathcal{C}$-backdoor B
Parameter:	$k = \|B\|$
Question:	Is F satisfiable?

Outline

Example



Incidence graph of the nested formula $F = \bigwedge_{i=1}^{8} c_i$ with

$$c_1 = t \vee \neg u, \quad c_2 = u \vee v \vee w, \quad c_3 = w \vee x, \quad c_4 = x \vee \neg y,$$
$$c_5 = y \vee \neg z, \quad c_6 = t \vee u \vee \neg w, \quad c_7 = \neg x \vee z, \quad c_8 = \neg t \vee w \vee x$$

Definition

Definition (Nested CNF formula [Knuth, 1990])

A CNF formula is nested if its variables can be linearly ordered such that there is no pair of clauses that straddle each other. A clause c straddles a clause c' if there are variables $x, y \in \text{var}(c)$ and $z \in \text{var}(c')$ such that $x < z < y$ in the linear ordering under consideration.

Known Results for Nested Formulas

- Polynomial time algorithm for SAT on nested formulas (linear if the ordering is given)
  [Knuth, 1990]
- Nested formulas have incidence treewidth ≤ 3
  [Biedl, Henderson, 2004]
- Thus, #SAT can be solved in polynomial time for nested formulas
  [Fischer, Makowsky, Ravve, 2008] [Samer, Szeider, 2010]
- A CNF formula F is nested iff the incidence graph of $F + u$ is planar, where u is a new universal clause
  [Kratochvíl, Křivánek, 1993]

Outline

Backdoor Detection and Evaluation

- Simple detection algorithm
  - for each k-subset B of variables and each assignment τ to B, check whether $F[\tau] \in$NESTED
  - run time: $\binom{n}{k} \cdot 2^k \cdot n^{O(1)}$
  - XP-algorithm
- Simple evaluation algorithm
  - for each assignment τ to B, check whether $F[\tau]$ is satisfiable
  - run time: $2^k \cdot n^{O(1)}$
  - FPT-algorithm

Strong NESTED-Backdoor Detection

Is strong NESTED-Backdoor Detection fixed-parameter tractable?

This question remains open.

To prove our main result, we FPT-approximate the strong NESTED-Backdoor Detection problem.

Strong NESTED-Backdoor Detection

Is strong NESTED-Backdoor Detection fixed-parameter tractable?

This question remains open.

To prove our main result, we FPT-approximate the strong NESTED-Backdoor Detection problem.

Strong NESTED-Backdoor Detection

Is strong NESTED-Backdoor Detection fixed-parameter tractable?

This question remains open.

To prove our main result, we FPT-approximate the strong NESTED-Backdoor Detection problem.

FPT-approximation

Theorem

There is an FPT *algorithm that either concludes that F has no strong* NESTED*-backdoor of size at most k or finds a strong* NESTED*-backdoor of F of size at most 2^k.*

Corollary (Main result)

SAT *and* #SAT *are fixed-parameter tractable parameterized by the size of a smallest strong backdoor set with respect to the base class of nested CNF formulas.*

FPT-approximation

Theorem

There is an FPT algorithm that either concludes that F has no strong NESTED*-backdoor of size at most k or finds a strong* NESTED*-backdoor of F of size at most 2^k.*

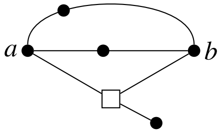
Corollary (Main result)

SAT *and* #SAT *are fixed-parameter tractable parameterized by the size of a smallest strong backdoor set with respect to the base class of nested CNF formulas.*

Outline of the FPT approximation algorithm

- If $\text{inc}(F)$ has "small" treewidth [Bodlaender, 1996]
  - express the problem in MSO_2
  - use Courcelle's theorem [Courcelle, 1990] [Arnborg, Lagergren, Seese, 1991]
- Otherwise
  - Compute a large grid minor [Robertson, Seymour, 1994] [Kawarabayashi, Kobayashi, Reed, 2012]
  - Compute a set S^* of $2^{O(k^{10})}$ variables such that every strong NESTED-backdoor contains at least one of these variables
  - For each $x \in S^*$, recurse on $F[x = 1]$ and $F[x = 0]$ with parameter $k - 1$
  - If, for some $x \in S^*$, both recursive calls return backdoors B_x and $B_{\neg x}$, then return $B_x \cup B_{\neg x} \cup \{x\}$
  - Otherwise, return NO

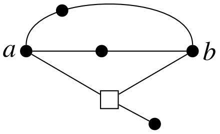
Obstruction



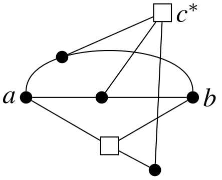
An obstruction.

The obstruction leads to a $K_{3,3}$-minor with the universal clause c^*.

Obstruction

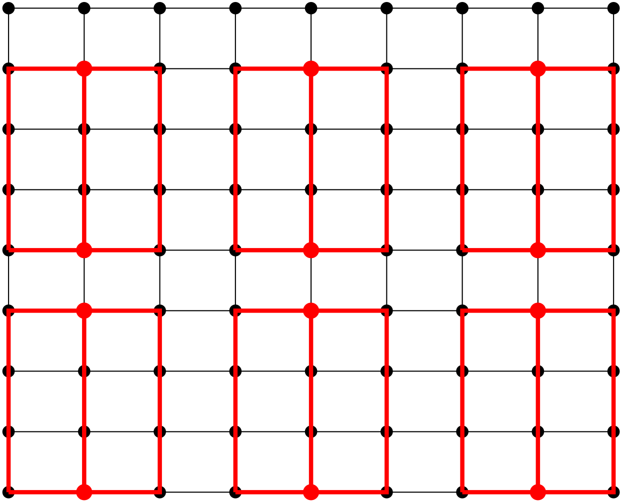


An obstruction.

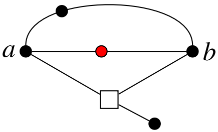


The obstruction leads to a $K_{3,3}$-minor with the universal clause c^*.

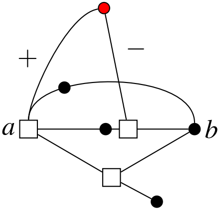
Using the Grid Minor



Internal and External Killers



An internal killer



An external killer

There are at most k internal killers.

Look closely at interaction of obstructions and external killers:
Either "sparse" interaction $\rightarrow$ no solution, or
or "dense" interaction $\rightarrow$ new obstructions emerge spanning external killers and known obstructions, forcing a backdoor to contain one variable among a few.

Our Results

Theorem

There is an FPT algorithm that either concludes that F has no strong NESTED*-backdoor of size at most k or finds a strong* NESTED*-backdoor of F of size at most 2^k.*

Corollary (Main result)

SAT *and* #SAT *are fixed-parameter tractable parameterized by the size of a smallest strong backdoor set with respect to the base class of nested CNF formulas.*

Other Base Classes

Base Class	*Weak* CNF	*Weak* r-CNF	*Strong* CNF	*Strong* r-CNF
HORN	W[2]-h	FPT	FPT	FPT
2CNF	W[2]-h	FPT	FPT	FPT
UP	W[P]-c	W[P]-c	W[P]-c	W[P]-c
RHORN	W[2]-h	W[2]-h	W[2]-h	open
CLU	W[2]-h	FPT	W[2]-h	FPT

The parameterized complexity of finding weak and strong backdoor sets of CNF formulas and r-CNF formulas, where $r \geq 3$ is a fixed integer.

Results by: [Nishimura, Ragde, Szeider, 2004] [Szeider, 2005] [Nishimura, Ragde, Szeider, 2007] [Gaspers, Szeider, 2012]
See [Gaspers, Szeider, 2012] for a survey.

Related Results

- FPT-approximation algorithm for detecting strong FOREST-backdoors [Gaspers, Szeider, ICALP 2012]
- FPT algorithm for detecting weak FOREST-backdoors for r-CNF formulas [Gaspers, Szeider, ICALP 2012]
- Weak FOREST-backdoor Detection is W[1]-hard for CNF formulas [Gaspers, Szeider, ICALP 2012]
- Faster and simpler randomized FPT algorithm for detecting weak FOREST-backdoors for r-CNF formulas [Fomin, Lokshtanov, Misra, Saurabh, unpublished]
- FPT-approximation algorithm for detecting strong TREEWIDTH_t-backdoors [Gaspers, Szeider, unpublished]

Note: $\text{TREEWIDTH}_1 = \text{FOREST} \subseteq \text{NESTED} \subseteq \text{TREEWIDTH}_3$

Thank you!

Questions?

Comments?