Proceedings Paper:

Fichte, JK, Ganian, R, Hecher, M et al. (2 more authors) (2023) Structure-Aware Lower Bounds and Broadening the Horizon of Tractability for QBF. In: 2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS). 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), 26-29 Jun 2023, Boston, USA. IEEE . ISBN 979-8-3503-3587-3

https://doi.org/10.1109/LICS56636.2023.10175675

Structure-Aware Lower Bounds and Broadening the Horizon of Tractability for QBF

***Abstract*—The QSAT problem, which asks to evaluate a quantified Boolean formula (QBF), is of fundamental interest in approximation, counting, decision, and probabilistic complexity and is also considered the prototypical PSPACE-complete problem. As such, it has previously been studied under various structural restrictions (parameters), most notably parameterizations of the primal graph representation of instances. Indeed, it is known that QSAT remains PSPACE-complete even when restricted to instances with constant treewidth of the primal graph, but the problem admits a double-exponential fixed-parameter algorithm parameterized by the vertex cover number (primal graph).**

However, prior works have left a gap in our understanding of the complexity of QSAT when viewed from the perspective of other natural representations of instances, most notably via incidence graphs. In this paper, we develop structure-aware reductions which allow us to obtain essentially tight lower bounds for highly restricted instances of QSAT, including instances whose incidence graphs have bounded treedepth or feedback vertex number. We complement these lower bounds with novel algorithms for QSAT which establish a nearly-complete picture of the problem's complexity under standard graph-theoretic parameterizations. We also show implications for other natural graph representations, and obtain novel upper as well as lower bounds for QSAT under more fine-grained parameterizations of the primal graph.

I. Introduction

The evaluation problem for quantified Boolean formulas (QSAT) is a natural generalization of the Boolean satisfiability problem (SAT) and among the most important problems in theoretical computer science, with applications in symbolic reasoning [1], [2], [3], [4], [5], [6], constraint satisfaction problems (CSP) [7], [8], [9], databases, and logic [10]. Input formulas in QSAT consist of a *(quantifier) prefix* and a *matrix*, which can be an arbitrary Boolean formula but is often assumed to be in *conjunctive normal form (CNF)*, e.g., converted by the classical Tseytin transformation [11]. QSAT is considered the archetypical representative of PSPACE-complete problems and has been extensively studied from the perspective of classical approximation [12], counting [13], decision [14], and probabilistic complexity [15], but also through the lens of parameterized complexity [16], [17].

The vast majority of parameterizations studied for QSAT rely on a suitable graph representation of the matrix; this is, in fact, similar to the situation for BOOLEAN SATISFIABILITY (SAT) [18], [19], [20], INTEGER LINEAR PROGRAMMING (ILP) [21], [22], [23], CONSTRAINT SATISFACTION (CSP) [9], [24], [25], and other fundamental problems. For QSAT, the most classical parameterization considered in the literature is the treewidth k of the *primal graph* representation of the formula's matrix in *conjunctive normal form (CNF)*. There, the complexity is well understood by now: The problem remains PSPACE-complete when parameterized by this parameter alone [16], [17], [26] even when restricted to decompositions which are paths, but can be solved in time $\mathsf{tow}(\ell, k) \cdot \mathrm{poly}(n)$[1] where ℓ is the quantifier depth of the formula's prefix and n is the number of variables of the formula. On a more positive note, parameterizing by the vertex cover number of the primal graph alone is known to yield a fixed-parameter algorithm for QSAT [27] that is double-exponential.

The ℓ-fold exponential gap in terms of parameter dependence between treewidth and vertex cover number raises the following question: what is the boundary of fixed-parameter tractability when dropping the quantifier depth ℓ as a parameter? In parameterized complexity, there is a whole hierarchy of structural parameters that are more restrictive than treewidth and less restrictive than vertex cover number, most prominently *treedepth* [28] and the *feedback vertex* and *edge numbers*[2]. However, there is an even larger gap: we know very little about the complexity-theoretic landscape of QSAT in the context of matrix representations other than the primal graph. The most prominent example of such a graph representation is the *incidence graph*, which has been extensively studied for SAT [18], [29], [19], CSP [25], [30], and ILP [22], [31], among others. The aforementioned hardness for QSAT carries over from primal treewidth to the treewidth of the incidence graph [17] and the problem is fixed-parameter tractable using quantifier depth plus treewidth of the incidence graph [32], but no other results for structural parameters of this graph were previously known.

A. Overview of Contributions

Inspired by the high-level approach used to obtain QSAT lower bounds for treewidth [26], in Section III we formalize a notion of *structure-aware* (SAW) reductions for QSAT. These reductions serve as a tool to

[1]The runtime is exponential in the treewidth k, where k is on top of a tower of iterated exponentials of height quantifier depth ℓ.

[2]The vertex or edge deletion distances to acyclicity, respectively.

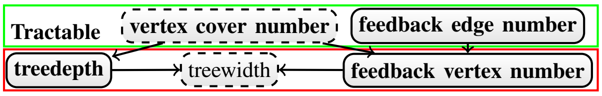

Fig. 1: Nearly-complete picture for QSAT and parameters on the incidence graph between vertex cover number and treewidth; directed arks indicate that the source parameter bounds the destination, see, e.g., [34], [35]. The red frame represents intractability with lower bounds essentially matching known upper bounds (ETH-tight for treewidth and feedback vertex number); the green frame indicates tractability (fpt) results. Bold-face text marks new results.

precisely demonstrate functional dependencies between parameters of the input instance and the reduced instance. Utilizing this notion of SAW reductions, we establish in Section IV tight lower bounds for highly restrictive classes of QSAT instances that have profound complexity-theoretic implications for three distinct representations of the matrix. These results essentially rule out efficient algorithms for treedepth and faster algorithms than the one for treewidth when using feedback vertex number. We highlight them below, followed by a separate discussion for each of the representation. Unless the *Exponential Time Hypothesis (ETH)* [33] fails:

1) QSAT cannot be solved faster than in $\mathsf{tow}(\ell', k) \cdot \mathrm{poly}(n)$ for ℓ' linear in the quantifier depth ℓ, where k is either the feedback vertex number ($\ell'{=}\ell$) or the treedepth of the incidence graph.
2) QSAT cannot be solved faster than in $\mathsf{tow}(\ell', k) \cdot \mathrm{poly}(n)$ for ℓ' linear in the quantifier depth ℓ, where k is either the feedback vertex number ($\ell'{=}\ell$) or the treedepth of the primal graph of formulas in combined conjunctive normal form (CNF) and *disjunctive normal form (DNF)*.
3) QSAT cannot be solved faster than in $\mathsf{tow}(\ell', k) \cdot \mathrm{poly}(n)$ for ℓ' linear in the quantifier depth ℓ, where k is either the feedback vertex number ($\ell'{=}\ell$) or the treedepth of the primal graph after deleting a single clause from the matrix.

1. Results for Incidence Graphs. Our two complexity-theoretic lower bounds identify that, with respect to the fundamental representation as incidence graph, the boundaries of intractability for QSAT lie significantly below treewidth. They also raise the question of whether we can obtain efficient algorithms for the problem using parameters which place stronger restrictions on the incidence graph. The two by far most natural structural graph parameters satisfying these properties are the mentioned vertex cover number and feedback edge number.

We complement our lower bounds with fixed-parameter algorithms for QSAT with respect to both parameters, which are provided in Section V. Thereby we establish a nearly-complete picture of the problem's complexity based on structure of the incidence graph, see Figure 1.

2. Implications for Primal Graphs of Combined Matrices. Previous complexity-theoretic studies of QSAT have typically assumed that the matrix is represented in CNF form, which admits standard graphical representations and may be obtained from an arbitrary formula by using the classical Tseytin transformation [11]. However, empirical evidence has shown that normal form transformations adversely affect the performance of QSAT solvers [36], and solvers now typically support more general input formats than CNF [37].

Given these developments, it is natural to consider the complexity of QSAT from the viewpoint of more general normal forms of the matrix which still admit suitable graph representations. An obvious step in this direction would be to combine CNF and DNF, i.e., consisting of a conjunction of a CNF and a DNF formula. This combined "CDNF" is used by backtracking search algorithms for QBF, since it is able to emulate forms of circuit-level reasoning while enjoying optimized data structures [38].

Since the CDNF is a strict generalization of the CNF, the lower bounds we established for the incidence graph of the CNF in Section IV immediately carry over. However, unlike in the CNF case, our reductions also directly rule out fixed-parameter tractability of QSAT with respect to both the treedepth and the feedback vertex number of primal graphs for matrices in CDNF.

3. Tightening the Gap on Primal Graphs. For classical CNF matrices, our SAW reductions of Section IV almost—but not quite—settle the aforementioned complexity-theoretic gap between the treewidth and vertex cover number of the primal graph. In particular, we prove that allowing the addition of a single clause to instances with bounded treedepth or feedback vertex number in the primal graph leads to intractability. Given this development, we view settling the parameterized complexity of QSAT with respect to these two parameters as the main open questions left in our understanding of the problem's complexity landscape.

As our last contribution, we obtain new fixed-parameter algorithms for QSAT with the aim of tightening this gap. First, we obtain a linear kernel (and hence also a fixed-parameter algorithm) for QSAT parameterized by the feedback edge number of the primal graph. Second, we establish the fixed-parameter tractability for the problem with respect to several relaxations of the vertex cover number that may be seen as "stepping stones" towards treedepth on primal graphs of CNFs. A more elaborated overview of our results is provided in Figure 2 (left).

B. Approach and Techniques

For establishing fine-grained lower bounds for parameters between treewidth and vertex cover on the graph representations above, we utilize the notion of structure-aware (SAW) reductions as visualized in Figure 2 (right). We develop concrete SAW reductions that are conceptually self-reductions from QSAT to QSAT, where we trade an exponential decrease of the parameters feedback vertex number or treedepth for an exponential increase of runtime dependency on the corresponding parameter. In order to obtain tight lower bounds that ideally match existing upper bounds (and rule out algorithms significantly better than the one for treewidth), one has to carefully carry out this trade-off such that the order of magnitude of the runtime dependency increase does not exceed the parameter decrease's magnitude. More precisely, our transformations reduce from QSAT using the respective parameter k and quantifier depth ℓ, to QSAT when parameterized by $\log(k)$ with quantifier depth $\ell{+}1$. By iterating this construction (see also Figure 2 (right)), we trade an i-fold exponential parameter decrease (from k to $\log^i(k)$) for a quantifier depth increase of i, which then, assuming ETH, results in a QBF that is $\ell{+}i$-fold exponential in $\log^i(k)$ to solve.

As a consequence of our reductions, we also obtain an interesting result for classical complexity: It turns out that *a single additional clause* is already responsible for intractability of QSAT on the well-known tractable fragment of 2-CNF formulas. More specifically, QSAT on 2-CNFs plus one clause with quantifier depth $\ell > 1$ is indeed $\Sigma^P_{\ell-1}$-complete, see Corollary XI.2.

Notably, the construction of our reductions also allows us to strengthen our established lower bounds to graph representations that are *purely restricted to variables of the innermost quantifier (block)*. This is a consequence of the fact that our concrete SAW reductions are carried out such that the majority of structural dependencies reside in the innermost quantifier block of the constructed instance. Further, the lower bounds even hold for parameters covering the vertex deletion distance to (almost) simple paths, as well as for restricted variants of treedepth. Both results are construction-specific consequences, but these findings are in fact significantly stronger than the lower bounds for feedback vertex set and treedepth, thereby providing deeper insights into the hardness of QSAT.

Our lower bound results using SAW reductions allow us to draw a rather comprehensive picture for the (fine-grained) parameterized complexity of the incidence graph by strengthening previous hardness results to much more restrictive parameters such as feedback vertex set and treedepth. We complement these negative results for the incidence graph by giving fpt-algorithms for CQSAT, i.e., QSAT restricted to formulas in CNF, both for vertex cover number and feedback edge number. Our main technical contribution here is a kernelization algorithm for feedback edge set for both the primal and incidence graph.

We then turn our attention towards solving CQSAT using structural restrictions of the primal graph. While we have to leave open whether CQSAT is fixed-parameter tractable parameterized by either the feedback vertex number or the treedepth of the primal graph, we are able to make some progress towards establishing fixed-parameter tractability for the latter. In particular, using novel insights into winning strategies of Hintikka games, we are able to obtain fixed-parameter algorithms for three variants of the so-called *c-deletion set* parameter, which is a parameter between vertex cover number and treedepth.

II. Preliminaries

We assume basics from graph theory, cf. [41], [42]. A graph $G = (V, E)$ is a *subgraph* of $G'{=}(V', E')$ if $V \subseteq V'$, $E \subseteq E'$. A *(connected) component* of a graph is a largest connected subgraph. A graph is *acyclic* if no subgraph forms a cycle. For a graph $G = (V, E)$ and a set $S \subseteq V$ ($D \subseteq E$) of vertices (edges), we define the *subtraction graph* obtained from G by $G - S := (V \setminus S, \{e \mid e \in E, e \cap S = \emptyset\})$ (by $G - D := (V \setminus \{v \mid v \in e, e \in D\}, E \setminus D)$). Further, the *union* of given graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is given by $G_1 \sqcup G_2 := (V_1 \cup V_2, E_1 \cup E_2)$. Expression $\mathsf{tow}(\ell, k)$ is a tower of exponentials of 2 of height ℓ with k on top.

Computational Complexity: We give a brief background on parameterized complexity [43], [44]. Let Σ and Σ' be two finite non-empty alphabets. A *parameterized problem* L is a subset of $\Sigma^* \times \mathbb{N}$ for some finite alphabet Σ. L is *fixed-parameter tractable (fpt)* if there exists a computable function f and an algorithm deciding whether $(\mathcal{I}, k) \in L$ in *fpt-time* $\mathcal{O}(f(k)\,\mathrm{poly}(\|\mathcal{I}\|))$, where $\|\mathcal{I}\|$ is the size of $\mathcal{I}$. Let $L \subseteq \Sigma^* \times \mathbb{N}$ and $L' \subseteq \Sigma'^* \times \mathbb{N}$ be two parameterized problems. A *polynomial-time parameterized-reduction* r, *pp-reduction* for short, from L to L' is a many-to-one reduction from $\Sigma^* \times \mathbb{N}$ to $\Sigma'^* \times \mathbb{N}$ such that $(\mathcal{I}, k) \in L$ if and only if $r(\mathcal{I}, k) = (\mathcal{I}', k') \in L'$ with $k' \leq p(k)$ for a fixed computable function $p : \mathbb{N} \to \mathbb{N}$ and r is computable in time $\mathcal{O}(\mathrm{poly}(\|\mathcal{I}\|))$. Parameter values are usually computed based on a structural property K of the instance, called *parameter*, e.g., size of a smallest feedback vertex set, or treewidth. Usually for algorithms we need structural representations instead of parameter values, i.e., a feedback vertex set or tree decomposition. Therefore, we let Γ be a finite non-empty alphabet and call $S \in \Gamma^*$ a *structural representation* of $\mathcal{I}$. Then, a *parameterization* κ for parameter K is a mapping $\kappa : \Gamma^* \to \mathbb{N}$ computing $k = \kappa(S)$ in polynomial time.

Quantified Boolean Formulas (QBFs): Boolean formulas are defined in the usual way [45], [46]; *literals* are variables or their negations. We let the *sign of a literal* l be defined by $\mathsf{sgn}(l) := 1$ if l is a variable and $\mathsf{sgn}(l) := 0$ otherwise. For a Boolean formula F, we denote by $\mathrm{var}(F)$ the set of variables of F. Logical operators $\to, \wedge, \vee, \neg$ refer to implication, conjunction, disjunction, and negation, respectively, as in the usual

Prefix^A	Matrix	Matrix-NF	Gph	Complexity	Ref
		CNF		PSPACE	[39]
qd ℓ		CNF		Σ_ℓ^P / Π_ℓ^P	[39]
qd ℓ		2,1-CDNF		$\Sigma_{\ell-1}^P$ / $\Pi_{\ell-1}^P$	Cor XI.2
	tw	CNF	P	PSPACE	[16], [17]
qd	tw	CNF	P, I	$\triangle^\dagger$ / $\triangledown^\ddagger$	Prop II.3 [40],[32] / [16], Prop II.4 [26]
qd	tw	m,1-CDNF^B	P	$\blacktriangle^\dagger$ / $\triangledown^\ddagger$	The III.2 / Prop II.4 [26]
qd	fvs	m,1-CDNF^B	P	$\blacktriangle^\dagger$ / $\blacktriangledown^\ddagger$	Cor III.4 / The IV.5
qd	fvs	CNF	I	$\blacktriangle^\dagger$, $\blacktriangledown^\ddagger$	Cor III.4 / Cor IV.6
qd	tdp	m,1-CDNF^B	P	$\blacktriangle^\dagger$ / $\blacktriangledown^{\sim\ddagger}$	Cor III.4 / The IV.7, Cor IV.8
qd	tdp	CNF	I	$\blacktriangle^\dagger$ / $\blacktriangledown^{\sim\ddagger}$	Cor III.4 / Cor IV.9
	fes	CNF	P	$\blacksquare$	The V.2
	fes	CNF	I	$\blacksquare$	The V.5
	dels	CNF	P	$\blacksquare^C$	The VI.3, The VI.5, The VI.7
	vc	m,1-CDNF^B	P	$\blacktriangle^{\dagger_2}$ / $\triangledown^{\ddagger_2}$	The III.1 / Prop II.5 [27]
	vc	CNF	I	$\blacktriangle^{\dagger_2}$	The V.1

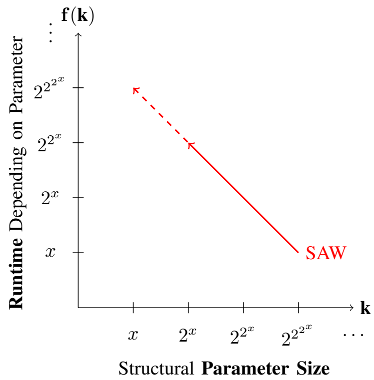

Fig. 2: (Left): Runtime bounds for QSAT on a QBF Q when parameterized by parameters listed in prefix and matrix. The triangles $\blacktriangle$ refer to established precise upper bounds and $\blacktriangledown$ to precise lower bounds. By $\triangledown$ and $\triangle$ we refer to previously known precise upper and lower bounds. The box $\blacksquare$ illustrates new fixed-parameter tractability results. The parameters are as follows: "qd" refers to the *quantifier depth*; "fvs" indicates the *feedback vertex number*; "tdp" indicates the *treedepth*; "fes"refers to the *feedback edge number*; and "dels" refers to the *size*+c for a *c-deletion set*. Our lower bounds are actually stronger and still hold for parameters when restricted to variables of the innermost quantifier block (see Appendix X). The runtime bounds are abbreviated by the marks where ℓ refers to the prefix and k to the parameterization of the matrix. Detailed results: †: $\mathsf{tow}(\ell, O(k)) \cdot \text{poly}(|\text{var}(Q)|)$; ‡: $\mathsf{tow}(\ell, o(k)) \cdot \text{poly}(|\text{var}(Q)|)$; $^{\sim\ddagger}$: $\mathsf{tow}(\ell, o(k-\ell)) \cdot \text{poly}(|\text{var}(Q)|)$ ^{†_2}: $2^{2^{\mathcal{O}(k)}} \cdot \text{poly}(|\text{var}(Q)|)$, for constant m: $2^{k^{\mathcal{O}(m)}} \cdot \text{poly}(|\text{var}(Q)|)$; ^{‡_2}: $2^{2^{o(k)}} \cdot \text{poly}(|\text{var}(Q)|)$; and ^{A}: $\exists$ odd/$\forall$ even. ^{B}: the lower bound already holds for 3,1-CDNF; and ^{C}: fpt under restrictions, parameterized by $|D|+c$ for a c-deletion set D. (Right): Structure-Aware (SAW) reductions for compensating exponential parameter decrease by increasing the parameter's runtime dependency.

meaning. A *term* or *clause* is a set S of literals which is interpreted as a conjunction or disjunction of literals, respectively.

We denote by $\text{var}(S)$ the set of variables appearing in S; without loss of generality we assume $|S| = |\text{var}(S)|$. A Boolean formula F is in *conjunctive normal form (CNF)* if F is a conjunction of clauses and F is in *disjunctive normal form (DNF)* if F is a disjunction of terms. In both cases, we identify F by its set of clauses or terms, respectively. A Boolean formula is in *d-CNF* or *d-DNF* if each set in F consists of at most d many literals. Let $\ell \geq 0$ be integer. A *quantified Boolean formula* Q is of the form $\mathcal{Q}.F$ for *prefix* $\mathcal{Q} = Q_1 V_1.Q_2 V_2.\cdots Q_\ell V_\ell$, where *quantifier* $Q_i \in \{\forall, \exists\}$ for $1 \leq i \leq \ell$ and $Q_j \neq Q_{j+1}$ for $1 \leq j \leq \ell - 1$; and where V_i are disjoint, non-empty sets of Boolean variables with $\text{var}(Q) := \text{var}(F) = \bigcup_{i=1}^{\ell} V_i$; and F is a Boolean formula. If F is in (c-)CNF, Q is a *(c-)CQBF*. We call ℓ the *quantifier depth* of Q and let $\mathsf{matr}(Q) := F$. Further, we denote the variables of Q by $\text{var}(Q) := \text{var}(F)$ and the existential (universal) variables by $\text{var}^\exists(Q)$ ($\text{var}^\forall(Q)$), defined by $\text{var}^\exists(Q) := \bigcup_{1 \leq i \leq \ell, Q_i = \exists} V_i$ ($\text{var}^\forall(Q) := \bigcup_{1 \leq i \leq \ell, Q_i = \forall} V_i$), respectively.

An *assignment* is a mapping $\alpha : X \to \{0, 1\}$ from a set X of variables. Given a Boolean formula F and an assignment α for $\text{var}(F)$. Then, for F in CNF, $F[\alpha]$ is a Boolean formula obtained by removing every $c \in F$ with $x \in c$ and $\neg x \in c$ if $\alpha(x) = 1$ and $\alpha(x) = 0$, respectively, and by removing from every remaining clause $c \in F$ literals x and $\neg x$ with $\alpha(x) = 0$ and $\alpha(x) = 1$, respectively. Analogously, for F in DNF values 0 and 1 are swapped. For a given QBF Q and an assignment α, $Q[\alpha]$ is a QBF obtained from Q, where variables x mapped by α are removed from preceding quantifiers accordingly, and $\mathsf{matr}(Q[\alpha]) = (\mathsf{matr}(Q))[\alpha]$.

A Boolean formula F *evaluates to true* (or *is satisfied*) if there exists an assignment α for $\text{var}(F)$ such that $F[\alpha] = \emptyset$ if F is in CNF or $F[\alpha] = \{\emptyset\}$ if F is in DNF. We say that then α *satisfies* F or α is a *satisfying assignment* of F. A QBF Q *evaluates to true (or is valid)* if $\ell = 0$ and $\mathsf{matr}(Q)$ evaluates to true under the empty assignment. Otherwise, i.e., if $\ell \neq 0$, we distinguish according to Q_1. If $Q_1 = \exists$, then Q evaluates to true if and only if there exists an assignment $\alpha : V_1 \to \{0, 1\}$ such that $Q[\alpha]$ evaluates to true. If $Q_1 = \forall$, then Q evaluates to true if for any assignment $\alpha : V_1 \to \{0, 1\}$, we have that $Q[\alpha]$ evaluates to true. We say that two QBFs are *equivalent* if one evaluates to true whenever the other does. Given a (C)QBF Q, the *evaluation problem* QSAT *(*CQSAT*)* of QBFs asks whether Q evaluates to true. Then, SAT is QSAT, but restricted to one $\exists$ quantifier. In general, QSAT is PSPACE-complete [46], [14], [39].

Example II.1. *Consider CQBF* $Q = \forall a, b.\exists c, d.C$*, where* $C := \{c_1, c_2, c_3, c_4\}$ *is a conjunction of clauses, with* $c_1 := \neg a \vee \neg b \vee c$*,* $c_2 := a \vee b \vee c$*,* $c_3 := \neg a \vee \neg c \vee d$*, and* $c_4 := a \vee \neg c \vee \neg d$*. Note that* $Q[\alpha]$ *is valid under any* $\alpha : \{a, b\} \to \{0, 1\}$*, which can be shown by giving an assignment* $\beta : \{c, d\} \to \{0, 1\}$ *for an arbitrary* α*. Concretely, let* $\beta(c) := 1$ *whenever* $\alpha(a) = \alpha(b)$ *and* $\beta(c) := 0$ *otherwise. Further,* $\beta(d) := 1$ *whenever* $\alpha(a) = \alpha(b) = 1$ *and* $\beta(d) := 0$ *otherwise. Indeed, for any such* α*, we have that* $C[\alpha][\beta] = \emptyset$ *and* $D[\alpha][\beta] = \{\emptyset\}$*. Consider, e.g.,* $\alpha = \{a \mapsto 0, b \mapsto 1\}$*, satisfying* c_1, c_2 *and* c_3*; then* c_4 *is satisfied by* β*.* ■

Extended Normalizations of the Matrix—Formulas in CDNF: Our investigations also consider a natural and more general *conjunctive/disjunctive normal form (CDNF)* for QBFs. A QBF Q, whose *innermost* quantifier is Q_ℓ, is *in CDNF*, whenever for a CNF C and DNF D, we have $\mathsf{matr}(Q) = C \wedge D$ if $Q_\ell = \exists$, and $\mathsf{matr}(Q) = D \vee C$ if $Q_\ell = \forall$. Naturally, we say that Q is in d*-CDNF* if C is in d-CNF and D is in d-DNF. Further, Q is in *d,1-CDNF* if C is in d-CNF and D is in 1-DNF (i.e., D is a long clause interpreted as a disjunction of singleton terms). Then, the problem QSAT_ℓ refers to QSAT when restricted to QBFs in CDNF and quantifier depth ℓ.

Graph Representations: In order to apply graph parameters to (Quantified) Boolean formulas, we need a graph representation. For a Boolean formula F in CNF or DNF we define the *primal graph* $G_F := (\text{var}(F), E)$ [18] over the variables of F, where two variables are adjoined by an edge, whenever they appear together in at least one clause or term of F, i.e., $E := \{\{x, y\} \mid f \in F, \{x, y\} \subseteq \text{var}(f), x \neq y\}$. The *incidence graph* $I_F := (\text{var}(F) \cup F, E')$ of F is over the variables and clauses (or terms) of F and $E' := \{\{f, x\} \mid f \in F, x \in \text{var}(f)\}$. For a QBF Q in CDNF with $\mathsf{matr}(Q) = C \wedge D$ or $\mathsf{matr}(Q) = D \vee C$, respectively, let the *primal graph* of Q be $G_Q := G_C \sqcup G_D$ and the *incidence graph* of Q be $I_Q := I_C \sqcup I_D$.

Example II.2. *Recall* Q *and* $C = \mathsf{matr}(Q)$ *from Example II.1; observe primal and incidence graphs* G_Q*,* I_Q *in Figure 3 (left,middle). Assume a QBF* Q' *in CDNF obtained from* Q*, where* $\mathsf{matr}(Q') := C \wedge D$ *with* D *being a disjunction of (singleton) terms, i.e.,* $D := \{\{b\}, \{\neg d\}\}$*. Note that by definition the (1-)DNF formula* D does not cause an additional edge *in the primal graph* $G_{Q'}$*, i.e., the graph is equivalent to the primal graph* G_C without D. *So,* G_Q *coincides with primal graph* $G_{Q'}$*; in general, for any QBF* Q *in CDNF,* $G_Q = G_{\bar{Q}}$*, using inverse* $\overline{Q}$*.* ■

Treewidth and Pathwidth: Let $G = (V, E)$ be a graph. A *tree decomposition (TD)* [47], [48] of graph G is a pair $\mathcal{T} = (T, \chi)$ where T is a tree, and χ is a mapping that assigns to each node t of T a set $\chi(t) \subseteq V$, called a *bag*, such that the following conditions hold:

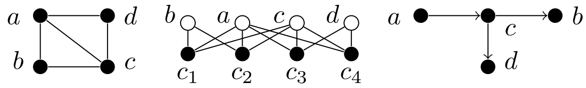

Fig. 3: (Left): Primal graph representation G_Q of the QBF Q of Example II.1. (Middle): Incidence graph I_Q of QBF Q. (Right): Treedepth decomposition of G_Q.

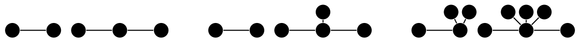

Fig. 4: Disjoint paths (left); half-ladder graph (middle); and caterpillar graph (right).

(i) $V = \bigcup_{t \text{ of } T} \chi(t)$ and $E \subseteq \bigcup_{t \text{ of } T} \{\{u, v\} \mid u, v \in \chi(t)\}$; and (ii) for each q, s, t, such that s lies on the path from q to t, we have $\chi(q) \cap \chi(t) \subseteq \chi(s)$. Then, $\mathsf{width}(\mathcal{T}) := \max_{t \text{ of } T} |\chi(t)| - 1$. The *treewidth* $\mathsf{tw}(G)$ of G is the minimum $\mathsf{width}(\mathcal{T})$ over all TDs $\mathcal{T}$ of G. For QSAT, we obtain the following tractability result.

Proposition II.3 (UB for Treewidth [40])**.** *Given any CQBF* Q *of quantifier depth* ℓ *with* $k = \mathsf{tw}(G_Q)$*. Then,* QSAT *on* Q *can be decided in time* $\mathsf{tow}(\ell, \mathcal{O}(k)) \cdot \text{poly}(|\text{var}(Q)|)$*.*

However, it is not expected that one can significantly improve this, since already for the weaker pathwidth there are limits. The *pathwidth* $\mathsf{pw}(G)$ of graph G is the minimum width over all TDs of G whose trees are paths.

Proposition II.4 (LB for Pathwidth [26])**.** *Given any CQBF* Q *of quantifier depth* ℓ *with* $k = \mathsf{pw}(G_Q)$*. Then, under ETH,* QSAT *on* Q *cannot be decided in time* $\mathsf{tow}(\ell, o(k)) \cdot \text{poly}(|\text{var}(Q)|)$*.*

Treedepth: Given a graph $G = (V, E)$. Then, a *treedepth decomposition* $T = (V, F)$ of G is a forest of rooted trees, where for every edge $\{u, v\} \in E$ we require that u is an ancestor or descendant of v in T. The *treedepth* $\mathsf{td}(G)$ of G is the smallest height among every treedepth decomposition of G, cf. Figure 3 (right).

Vertex Cover Number: Given a graph $G = (V, E)$. Then, a set $S \subseteq V$ of vertices is a *vertex cover (of G)* if for every edge $e \in E$ we have that $e \cap S \neq \emptyset$. Further, we define the *vertex cover number* of a graph G to be the smallest size among every vertex cover of G. Interestingly, QSAT is tractable when parameterized by this number.

Proposition II.5 (UB for Vertex Cover Number [27])**.** *Given any CQBF* Q *of* QSAT *with* k *being the vertex cover number of* G_Q*. Then, the validity of* Q *can be decided in time* $2^{2^{\mathcal{O}(k)}} \cdot \text{poly}(|\text{var}(Q)|)$ *(*$2^{\mathcal{O}(k^3)} \cdot \text{poly}(|\text{var}(Q)|)$ *for* $\mathsf{matr}(Q)$ *in 3-CNF).*

Feedback Sets and Distance Measures: Lower bound results for QSAT parameterized by treewidth (pathwidth) or vertex cover number motivates other parameters.

Let $G = (V, E)$ be a graph. Then, a set $S \subseteq V$ of vertices is called a *feedback vertex set (FVS)* of G if $G - S$

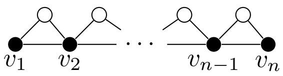

Fig. 5: Graph of pathwidth 2 with FV number in $\mathcal{O}(n)$.

is an acyclic graph, and the *feedback vertex number* (of G) refers to the smallest size among all feedback vertex sets of G. Further, S is referred to by *distance set to half-ladder* if $G - S$ is a *half-ladder (graph)*, consisting of (vertex) disjoint paths such that additionally each vertex might be adjacent to *one* fresh vertex. If we allow more than one such fresh vertex, we call the graph a *caterpillar*, cf. Figure 4. The smallest $k = |S|$ among these distance sets S is the *distance* (to the corresponding graph class). We say S is a *c-deletion set*, for some integer c, if every component of $G - S$ has at most c vertices. A set $D \subseteq E$ is a *feedback edge set (FES)* for G if $G - D$ is acyclic.

We utilize these sets S, D *for a QBF* Q, where $G = G_Q$. Then, S is *sparse* if for every two distinct vertices u, v of $G_Q - S$ there is at most one clause or term f of $\mathsf{matr}(Q)$ with $u, v \in \mathrm{var}(f)$.

Example II.6. *Recall QBF Q from Example II.1 and observe that the feedback vertex number is 1, e.g., $\{a\}$ is a FVS of G_Q as well as a distance set to half-ladder of G_Q. However, the sparse feedback vertex number of Q is 2, since no single vertex can be removed from G_Q such that each edge corresponds to at most one clause of $\mathsf{matr}(Q)$. Set $\{a, c\}$ is a sparse FVS of Q since $G_Q - \{a, c\}$ is disconnected. While for G_Q pathwidth is identical to the sparse FV number, the graph of Figure 5 has pathwidth 2, but admits only a large FVS, e.g., all white nodes.* ■

Inspired by related and more general works on parameter hierarchies [34], [35], we obtain a hierarchy of parameters for QBFs: Figure 6 depicts parameters, where a directed arc from parameter p_1 to parameter p_2 indicates that p_1 is *weaker* than p_2, i.e., p_2 is bounded by $\mathcal{O}(p_1)$. Consequently, lower bounds for the weaker (linearly smaller) parameter p_1 form *stronger results* and automatically carry over to the stronger parameter p_2.

Example II.7. *Observe that already for a CQBF Q, the feedback vertex number k_G of G_Q and the feedback vertex number k_I of I_Q are incomparable, cf., Figure 6. It is easy to see that $k_G \ll k_I$ by constructing an instance with one large clause. However, there are also cases where $k_I \gg k_G$: One can construct pairs of variables where each pair appears in (at least) two clauses of size 2, i.e., each pair is involved in a cycle in G_I. Then, k_G is zero, but k_I amounts to the number of pairs.*

III. Structure-Aware (SAW) Reductions

Recall the gap between runtimes for QSAT using treewidth (pathwidth, cf., Proposition II.4) and QSAT when parameterized by vertex cover number (see Proposition II.5). Interestingly, runtime bounds for QSAT and vertex cover number on CNFs also hold on CDNFs.

Theorem III.1 (UB for QSAT$_\ell$ and Vertex Cover Number, $\star$[3])**.** *There is an algorithm that, given a QBF Q in CDNF with vertex cover number k of G_Q, decides whether Q is true in time $2^{2^{\mathcal{O}(k)}} \cdot \mathrm{poly}(|\mathrm{var}(Q)|)$. If Q is in d-CDNF, the algorithm runs in time $2^{k^{\mathcal{O}(d)}} \cdot \mathrm{poly}(|\mathrm{var}(Q|))$.*

Proof (Idea). The result can be established by enhancing a DPLL-style backtracking algorithm with formula caching [49]. The number of subformulas of the matrix that can be obtained by assigning variables can be bounded by a function that is linear in the number of variables and only exponential in the size of the vertex cover. This upper bounds the size of the search tree. □

Similarly, the known runtime result for treewidth (Proposition II.3) carries over to CDNFs.

Theorem III.2 (UB for QSAT$_\ell$ and Treewidth)**.** *Given any QBF Q in CDNF of quantifier depth ℓ with $k = \mathsf{tw}(G_Q)$. Then,* QSAT$_\ell$ *on Q can be decided in time $\mathsf{tow}(\ell, \mathcal{O}(k)) \cdot \mathrm{poly}(|\mathrm{var}(Q)|)$.*

We show this result by relying on the following concept of structure-aware reductions. These reductions will be a key component of the constructions for the new lower bound results of this paper. They provide a constructive way of utilizing an actual structural representation of the instance, thereby precisely bounding the parameter increase (decrease) in terms of the representation.

Definition III.3 (Structure-Aware (SAW) Reduction)**.** *Let Σ, Σ', Γ, and Γ' be alphabets, $\mathcal{P} \subseteq \Sigma^* \times \mathbb{N}$ and $\mathcal{P}' \subseteq \Sigma'^* \times \mathbb{N}$ parameterized problems with parameterizations κ, κ', and f a computable function. An f-structure-aware reduction $\mathcal{R}$ from problem $\mathcal{P}$ to $\mathcal{P}'$ is a mapping from $\Sigma^* \times \Gamma^*$ to $\Sigma'^* \times \Gamma'^*$ such that for all $(\mathcal{I}, S) \in \Sigma^* \times \Gamma^*$ where $(\mathcal{I}', S') = \mathcal{R}(\mathcal{I}, S)$, we have that $(I, \kappa(S)) \mapsto (I', \kappa'(S'))$ is a pp-reduction s.t. (i) there is a polynomial-time function g with $S' = g(S)$* (functional dependency) *and (ii) $\kappa'(S') \leq \mathcal{O}(f(\kappa(S)))$* ($f$-boundedness).

The definition of SAW reductions serves the following purposes. First, such a reduction always provides a structural representation of the reduced instance, whereby (i) the functional dependency immediately gives insights into how such a representation can be obtained. Further, the property (ii) f-boundedness ensures that the resulting parameter of the reduced instance fulfills precise guarantees, which will be essential for the next subsection.

To demonstrate these reductions, we briefly explain the arcs of Figure 6. Interestingly, any of the arcs of Figure 6, except the ones from bold-face parameters to pathwidth or treewidth can be shown by the trivial linear-SAW

[3]Statements marked with a star ("$\star$") are proven in the appendix.

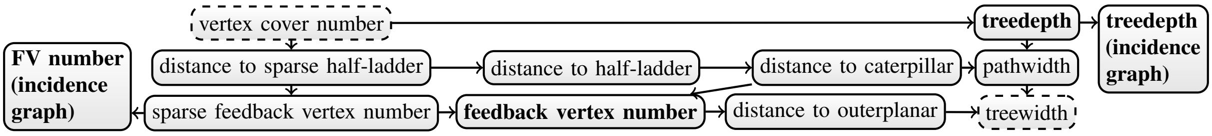

Fig. 6: Parameters for QBFs of the primal graph (incidence graph) between vertex cover number and treewidth. A directed arc indicates the source being weaker than the destination, i.e., the destination is linearly bounded by the source. Bold-face parameters mark new lower bound results *between vertex cover number and treewidth* for QSAT_ℓ.

reduction that takes a QBF Q of QSAT_ℓ and a structural representation S of the respective parameter and returns (Q, S). Indeed, e.g., any vertex cover S is a distance set to half-ladder and for any path decomposition S it holds that it is a tree decomposition. Further, any distance set S to outerplanar can be turned into a TD S', since each outerplanar graph [50] has a TD $\mathcal{T}$ of width at most 2 and we obtain TD S' by adding S to every bag of $\mathcal{T}$.

With these SAW reductions at hand, one can easily establish Theorem III.2. Since the treewidth parameter is linearly bounded by both feedback vertex size and treedepth, we instantly obtain the following results.

Corollary III.4 (UB for QSAT_ℓ and Feedback Vertex Number/Treedepth)**.** *Given any QBF Q in CDNF of quantifier depth ℓ with k being the feedback vertex number or treedepth of G_Q. Then, QSAT_ℓ on Q can be decided in time* $\mathsf{tow}(\ell, \mathcal{O}(k)) \cdot \mathrm{poly}(|\mathrm{var}(Q)|)$.

IV. Lower Bounds via SAW Reductions

In order to establish conditional lower bounds for parameterized problems, one might reduce from SAT on 3-CNFs and then directly apply the widely believed exponential time hypothesis (ETH) [33]. Indeed, many results have been shown, where the parameter of the reduced instance depends on the number of variables of the Boolean formula, e.g., [51], [27], [52], immediately followed by applying ETH. Oftentimes consequences of the ETH are sufficient, claiming that SAT on 3-CNFs cannot be solved in time $2^{o(k)} \cdot \mathrm{poly}(n)$, where k is a parameter of the instance and n is the variable number.

Having established the concept of structure-aware (SAW) reductions, we apply this type of reductions as a precise tool for generalizing the lower bound result of Proposition II.4. More specifically, the next subsection focuses on defining a self-reduction from QSAT to QSAT, where we trade an increase of quantifier depth for an exponential decrease in the parameter of interest. This is done in such a way that we are able to show lower bounds matching their upper bounds when assuming ETH.

In order to find suitable candidate parameters, recall that for the vertex cover number k, the problem QSAT can be solved in double-exponential runtime in k, regardless of quantifier depth. However for the treewidth (or pathwidth) this is not the case, since for quantifier depth ℓ, one requires a runtime that is ℓ-fold exponential in the pathwidth, cf. Proposition II.4. This motivates our quest to investigate suitable parameters that are "between" vertex cover number and pathwidth. In Section IV-A, we show that the (sparse) feedback vertex number is insufficient as well, i.e., we obtain ETH-tight lower bounds that match the upper bounds of Corollary III.4 for this parameter. Section IV-B adapts the reduction, thereby providing deeper insights into hardness for treedepth. Further, this section outlines an ETH-tight lower bound for treedepth, similar to Corollary III.4, for instances of high treewidth.

A. Tight QBF Lower Bound for Feedback Vertex Number

The overall approach proceeds via SAW reductions as follows. We assume an instance Q of QSAT_ℓ and a sparse feedback vertex set S such that $|S| = k$. Then, we devise a $\log(k)$-SAW reduction $\mathcal{R}$, constructing an equivalent QBF Q' such that Q' has a sparse feedback vertex set S' of size $\mathcal{O}(\log(k))$.

Without loss of generality, we restrict ourselves to the case where the innermost quantifier of the QBF Q is $\exists$, as one can easily adapt for the other case or solve the inverse problem and invert the result in constant time. Further, we assume that the first quantifier of Q is $\exists$ as well. Let $Q = \exists V_1.\forall V_2.\cdots\exists V_\ell.C \wedge D$ be such a QBF that admits a sparse feedback vertex set S with $k = |S|$. For the purpose of our lower bound, we actually prove a *stronger result in the more restricted 3,1-CDNF form*, where we assume C in 3-CNF and D in 1-DNF, i.e., $G_Q = G_C$ as discussed in Example II.2. Finally, we assume that each $c_i \in C$ consists of exactly three literals; however, the reduction works with individual smaller clause sizes. The reduced instance Q' and sparse feedback vertex set S' of Q' that is obtained by the SAW reduction, uses the additional quantifier block in order to "unfold" S' (i.e., reconstruct an assignment of S).

Auxiliary Variables: In order to construct Q', we require the following *additional (auxiliary) variables*. First, we use pointer or *index variables* that are used to address precisely *one element* of S. In order to address 3 elements of S for the evaluation of a 3-CNF (3-DNF) formula, we require three of those indices. These index variables are of the form $\mathit{VarIdxs} := \{idx_j^1, \ldots, idx_j^{\lceil\log(|S|)\rceil} \mid 1 \leq j \leq 3\}$ and, intuitively, for each of the three indices these allow us to "address" each of the k many elements of S via a specific assignment of $\lceil \log(k) \rceil$ many Boolean variables. These $2^{\lceil \log(k) \rceil}$ many combinations of variables per index j are sufficient to address any of the k elements of S. To this end, we assign each element $x \in S$ and each $1 \leq j \leq 3$ a set consisting of an arbitrary, but fixed and *unique combination of literals* over the index variables $idx_j^1, \dots, idx_j^{\lceil \log(|S|) \rceil}$, denoted by $[\![x]\!]_j$.

Further, for each clause $c_i \in C$ with $c_i = \{l_1, l_2, l_3\}$ we assume an arbitrary ordering among the literals of c_i and write $\mathsf{lit}(c_i, j) := l_j$ for the *j-th literal of* c_i ($1 \leq j \leq 3$). We also require three Boolean variables val_1, val_2, val_3, where val_j captures a *truth (index) value* for the element of S that is addressed via the variables for the j-th index. These variables are referred to by $VarVals := \{val_1, val_2, val_3\}$. Finally, we use one variable to store whether D is satisfied as well as $|C|$ many auxiliary variables that indicate whether a clause $c \in C$ is satisfied. These variables are addressed by the set $VarSat := \{sat, sat_1, \dots, sat_{|C|}\}$ of *satisfiability variables*, where we assume clauses $C = \{c_1, \dots, c_{|C|}\}$ are ordered according to some fixed total ordering.

The Reduction: The reduction $\mathcal{R}$ takes Q and S and constructs an instance Q' as well as a sparse feedback vertex set S' of Q'. The QBF Q' is of the form $Q' := \exists V_1. \forall V_2. \cdots \exists V_\ell. \forall VarIdxs, VarVals, VarSat. C' \vee D'$, where C' is in DNF, defined as a disjunction of terms:

$$x \wedge \bigwedge_{b \in [\![x]\!]_j} b \wedge \neg val_j \quad \text{for each } x \in S, 1 \leq j \leq 3 \tag{1}$$

$$\neg x \wedge \bigwedge_{b \in [\![x]\!]_j} b \wedge val_j \quad \text{for each } x \in S, 1 \leq j \leq 3 \tag{2}$$

$$sat_i \wedge l \quad \text{for each } c_i \in C, 1 \leq j \leq 3 \text{ with } \mathsf{lit}(c_i, j) = l, \mathrm{var}(l) \in \mathrm{var}(C) \setminus S \tag{3}$$

$$sat_i \wedge \neg b \quad \text{for each } c_i \in C, 1 \leq j \leq 3, x \in S, b \in [\![x]\!]_j \text{ with } \mathrm{var}(\mathsf{lit}(c_i, j)) = x \tag{4}$$

$$sat_i \wedge val_j \quad \text{for each } c_i \in C, 1 \leq j \leq 3, x \in S \text{ with } \mathsf{lit}(c_i, j) = x \tag{5}$$

$$sat_i \wedge \neg val_j \quad \text{for each } c_i \in C, 1 \leq j \leq 3, x \in S \text{ with } \mathsf{lit}(c_i, j) = \neg x \tag{6}$$

$$sat \wedge l \quad \text{for each } \{l\} \in D \tag{7}$$

Additionally, we define D' in 1-CNF, which is a conjunction of the following singletons.

$$\neg sat_i \quad \text{for each } 1 \leq i \leq |C| \tag{8}$$

$$\neg sat \tag{9}$$

Observe that the fresh auxiliary variables appear under the innermost universal quantifier of Q'. So, intuitively, Formulas (1) ensure that whenever some $x \in S$ is set to 1 and the j-th index targets x, that we then "skip" the corresponding assignment if val_j is set to 0. This is similar to Formulas (2) for the case $x \in S$ is set to false, ensuring that for the remaining formulas of Q' whenever the j-th index targets some $x \in S$, the corresponding value val_j agrees with the implicit assignment of x.

Formulas (3)–(6) are used to check that the clauses $c_i \in C$ are satisfied. Intuitively, the variables sat_i serve as switches that require clause c_i to be satisfied if set to true. Formulas (3) ensure that c_i is satisfied whenever a literal $l \in c_i$, whose variable $\mathrm{var}(l)$ is not in S, is assigned true. For literals $l \in c_i$ whose variables $\mathrm{var}(l)$ are in S, Formulas (4) evaluate to true if the corresponding j-th index (j such that $l = \mathsf{lit}(c_i, j)$) does not target $\mathrm{var}(l)$, or one of Formulas (5) and (6) is true if the targeted literal is true. Finally, one of Formulas (7) evaluates to true if D is true, similarly to Formulas (3). Observe that since the $VarSat$ variables are universally quantified, multiple sat_i variables might be set to true. Intuitively, this makes it "easier" to satisfy some term among Formulas (3)–(7), since it is sufficient for one of the clauses c_i to be satisfied. The only problematic assignment of $VarSat$ variables is the one where both sat as well as all the sat_i variables are set to false. This is prevented by 1-CNF formula D', i.e., Formulas (8),(9).

Example XI.1 ($\star$) illustrates $\mathcal{R}$ on a specific formula.

Structure-Awareness: Besides Q', reduction $\mathcal{R}$ above further gives rise to the sparse feedback vertex set of Q' defined by $S' := VarIdxs \cup VarVals \cup \{sat\}$. Indeed, the size of S' compared to $|S|$ is exponentially smaller and therefore $\mathcal{R}$ is indeed a structure-aware reduction. The reduction and the relations between Q and Q', as well as S and S' are visualized in Figure 7. Formally, we obtain the following result stating that $\mathcal{R}$ is indeed a SAW reduction for sparse feedback vertex sets.

Lemma IV.1 (Decrease Feedback Vertex Number, $\star$)**.** *Given QBF Q in 3,1-CDNF and a sparse feedback vertex set S of Q, $\mathcal{R}$ constructs QBF Q' with sparse feedback vertex set S' of Q' such that $|S'|$ is in* $\mathcal{O}(\log(|S|))$.

Towards 3-DNF of C': Observe that the formula C' generated by the reduction $\mathcal{R}$ is almost in 3-DNF. The only formulas that are not already in the required format are Formulas (1) and (2). It is easy to observe that, however, even those formulas can be transformed such that only at most 3 literals per term are used. To this end, one needs to introduce additional auxiliary variables (that are added to the innermost $\forall$ quantifier). Indeed, a straightforward transformation recursively splits Formulas (1) and (2) into two terms, where the first term consists of two literals and a new auxiliary variable v that is added (positively), and the second term consists of the remaining literals of the term and $\neg v$. In turn, each term has at most two new auxiliary variables and one only needs to take care that the resulting term that contains $x \in S$ or $\neg x \in S$ does not use two of these auxiliary variables (preventing cycles in the resulting primal graph).

Runtime and Correctness: Next, we show runtime and correctness of $\mathcal{R}$, followed by main results.

Theorem IV.2 (Runtime, $\star$)**.** *For a QBF Q in 3,1-CDNF with $\mathsf{matr}(Q) = C \wedge D$ and set $S \subseteq \mathrm{var}(Q)$ of variables of Q, $\mathcal{R}$ runs in time* $\mathcal{O}(\lceil \log(|S|+1) \rceil \cdot (|S|+|C|)+|D|)$.

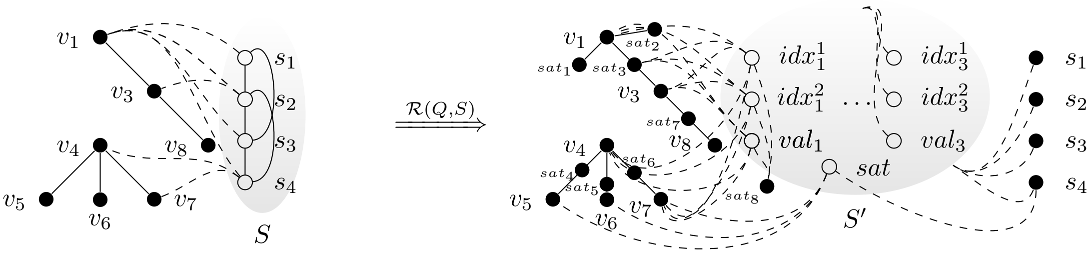

Fig. 7: Structure-aware reduction $\mathcal{R}$ for some QBF Q; dashed lines show potentially dense graph parts. (Left): A primal graph G_Q together with a sparse feedback vertex set S that connects G_Q. (Right): Corresponding primal graph $G_{Q'}$ (simplified) and the resulting sparse feedback vertex set S', where Q' and S' are obtained by $\mathcal{R}(Q, S)$. The illustration depicts three different kind of clauses of $\mathsf{matr}(Q)$: type (i) using only variables in S, like sat_8; type (ii): using two variables in S, like sat_1, sat_2, and type (iii) using only one variable in S, like $sat_3, sat_4, sat_5, sat_6, sat_7$.

Theorem IV.3 (Correctness, $\star$). *Given a QBF Q in 3,1-CDNF and a set $S \subseteq \mathrm{var}(Q)$ of variables of Q, reduction $\mathcal{R}$ computes an instance Q' that is equivalent to Q. In fact, any assignment α to variables of $\mathsf{matr}(Q)$ satisfies $\mathsf{matr}(Q)$ iff every extension α' of α to variables $VarIdxs \cup VarVals \cup VarSat$ satisfies $\mathsf{matr}(Q')$.*

QSAT is well known to be polynomial-time tractable when restricted to 2-CNF formulas [53]. As an application of our reduction $\mathcal{R}$, we now observe that allowing a single clause of arbitrary length already leads to intractability.

Corollary IV.4 ($\star$). *Problem* QSAT *over a QBF $Q{=}Q_1V_1 \cdots Q_\ell V_\ell.C \wedge D$ of quantifier depth $\ell \geq 2$ with $Q_\ell = \exists$, C being in 2-CNF, and D being in 1-DNF, is $\Sigma^{\mathrm{P}}_{\ell-1}$-complete (if $Q_1 = \exists$, ℓ odd) and $\Pi^{\mathrm{P}}_{\ell-1}$-complete (if $Q_1 = \forall$, ℓ even).*

A similar result can be obtained with long terms when the innermost quantifier is universal.

Lower Bound Result: Having established structure-awareness, runtime, as well as correctness of the reduction $\mathcal{R}$ above, we proceed with the lower bound results.

Theorem IV.5 (LB for Sparse Feedback Vertex Set, $\star$). *Given an arbitrary QBF Q in CDNF of quantifier depth ℓ and a minimum sparse feedback vertex set S of Q with $k = |S|$. Then, under ETH,* QSAT_ℓ *on Q cannot be decided in time $\mathsf{tow}(\ell, o(k)) \cdot \mathrm{poly}(|\mathrm{var}(Q)|)$.*

As a consequence, we obtain the following result.

Corollary IV.6 (LB for Incidence Feedback Vertex Set, $\star$). *Given an arbitrary QBF Q with $F = \mathsf{matr}(Q)$ in CNF (DNF) such that the innermost quantifier Q_ℓ of Q is $Q_\ell = \exists$ ($Q_\ell = \forall$) with the feedback vertex number of I_F being k. Then, under ETH,* QSAT_ℓ *on Q cannot be decided in time $\mathsf{tow}(\ell, o(k)) \cdot \mathrm{poly}(|\mathrm{var}(Q)|)$.*

The lower bound for the (sparse) feedback vertex number carries over to even more restrictive parameters, see Appendix X. Further, the lower bound holds still holds when restricting feedback vertex sets to variables of the innermost quantifier block, see Corollary X.3.

B. Hardness Insights & New Lower Bounds for Treedepth

So far, we discussed in Section IV-A why Corollary III.4 cannot be significantly improved for feedback vertex number. In this section, we provide a hardness result in the form of a conditional lower bound for the parameter treedepth. Notably, our approach for treedepth also involves a SAW reduction, where it turns out that we can even reuse major parts of reduction $\mathcal{R}$ as defined by Formulas (1)–(9). Let $Q = \exists V_1.\forall V_2.\cdots \exists V_\ell.C \wedge D$ be a QBF in CDNF and T be a treedepth decomposition of G_Q that consists of a path S of height h, where each element of the path might be connected to a tree of constant height. The result of applying $\mathcal{R}$ on Q and S is visualized in Figure 8. The final normalization step (from DNF to 3-DNF) results in multiple paths of length $\mathcal{O}(\log(h))$ due to additional auxiliary variables when normalizing Formulas (1) and (2). These paths do not increase the size of a sparse feedback vertex set and could previously be ignored. But reducing the treedepth requires compressing *each* of these paths, and so we must turn the reduction $\mathcal{R}$ that takes a single set S as an argument into a SAW reduction $\mathcal{R}_{\mathsf{td}}$ dealing with multiple paths simultaneously. Formal details of $\mathcal{R}_{\mathsf{td}}$ are given in Appendix XI-B.

Using $\mathcal{R}_{\mathsf{td}}$, we obtain the following lower bound for treedepth k, which yields an ETH-tight lower bound if $k \in \mathcal{O}(\ell)$ for quantifier depth ℓ, see Corollary III.4.

Theorem IV.7 (LB for Treedepth Decompositions, $\star$). *Given an arbitrary QBF Q in CDNF of quantifier depth ℓ and a treedepth decomposition T of G_Q of height $k = \mathsf{td}(G_Q)$. Then, under ETH,* QSAT_ℓ *on Q cannot be decided in time $\mathsf{tow}(\ell, o(k - \ell)) \cdot \mathrm{poly}(|\mathrm{var}(Q)|)$.*

We show that this result still implies a hierarchy of runtimes under ETH, where the tower height depends linearly on the quantifier depth of the QBF.

Corollary IV.8 (LB for Treedepth, $\star$). *Given a QBF Q in 3,1-CDNF of quantifier depth ℓ such that $k = \mathsf{td}(G_Q)$. Under ETH there exists $\ell' \in \Theta(\ell)$ such that* QSAT_ℓ *on Q cannot be decided in time $\mathsf{tow}(\ell', o(k)) \cdot \mathrm{poly}(|\mathrm{var}(Q)|)$.*

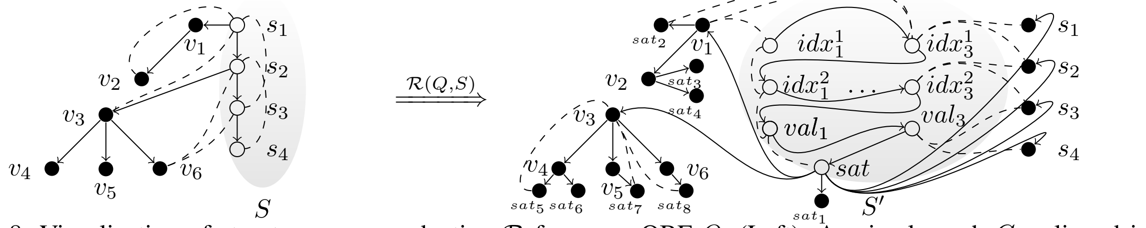

Fig. 8: Visualization of structure-aware reduction $\mathcal{R}$ for some QBF Q. (Left): A primal graph G_Q aligned in a treedepth decomposition T of depth h, where S is a path in T of height $\mathcal{O}(h)$; dashed lines show potential edges between ancestors and descendants. (Right): The corresponding primal graph $G_{Q'}$ (simplified), aligned in a treedepth decomposition T' and a corresponding path S' in T' of height $\mathcal{O}(\log(h))$; Q' and S' are obtained by $\mathcal{R}(Q,S)$.

Corollary IV.9 (LB for Incidence Treedepth, $\star$). *Given a QBF Q with $F = \mathsf{matr}(Q)$ in CNF (DNF) such that the innermost quantifier Q_ℓ of Q is $Q_\ell = \exists$ ($Q_\ell = \forall$) and $k = \mathsf{td}(I_Q)$. Then, under ETH,* QSAT$_\ell$ *on Q cannot be decided in time* $\mathsf{tow}(\ell, o(k-\ell)) \cdot \mathrm{poly}(|\mathrm{var}(Q)|)$.

Note that this corollary immediately yields a result similar to Corollary IV.8 for the incidence graph. Further, similar to consequences of the previous section, see Appendix X, we can obtain stronger results by restricting treedepth decompositions to variables of the innermost quantifier.

V. Algorithms Using Vertex Cover and FES

Our results from the previous section already provide a rather comprehensive picture of the (fine-grained) parameterized complexity of CQSAT, when considering many of the most prominent structural parameters on the incidence graph. In particular, they rule our fixed-parameter tractability of treedepth and feedback vertex set. In this section, we will complement this picture for the incidence graph by giving fpt-algorithms for CQSAT parameterized by the vertex cover number as well as the feedback edge set number. We start with our algorithm for the vertex cover number, which essentially follows from the simple observation that formulas with a small vertex cover number cannot have too many distinct clauses together with the well-known result that CQSAT is fpt parameterized by the number of clauses [54].

Theorem V.1 ($\star$). *Given any CQBF Q of* QSAT *with k being the vertex cover number of I_Q. Then, the validity of Q can be decided in time* $1.709^{3^k} \cdot \mathrm{poly}(|\mathrm{var}(Q)|)$.

Note that tractability for the vertex cover number of the incidence graph does not immediately carry over to the primal graph and therefore neither Proposition II.5 nor Theorem III.1 are a direct consequence of Theorem V.1; indeed a small vertex cover number of the primal graph still allows for an arbitrary number of distinct clauses.

We are now ready to provide our algorithm for the feedback edge number of the incidence graph. Interestingly and in contrast to vertex cover number, the parameterized complexity of CQSAT for the feedback edge number has been open even for the primal graph. While the FEN of the primal graph and the incidence graph are again orthogonal parameters (consider, e.g., two variables that occur together in more than one clause), we will show that the algorithm for the incidence graph can essentially be obtained using the techniques developed for the primal graph. We will therefore start by giving our result for the FEN of the primal graph, which also constitutes the main technical contribution of this section. We establish the result by proving existance of a kernelization algorithm.

Theorem V.2 ($\star$). *Let Q be a* CQBF*. In polynomial time, we can construct an equivalent* CQBF *with at most $12k-8$ variables and at most $10k-9+3^{\lfloor(\sqrt{24k+1}+1)/2\rfloor}$ clauses, where k is the feedback edge number of G_Q.*

The main ideas behind the kernelization are as follows. Given Q, we first compute a smallest FES D of primal graph $G{=}G_Q$ in polynomial time. Then, graph $H{=}G{-}D$ is a (spanning) forest of G. We introduce a series of reduction rules that allow us to reduce the size of Q and H. We start by observing that we can remove *unit clauses*, i.e., clauses containing only one literal, and *pure literals*, i.e., variables that either only occur positively or only negatively in Q. We then consider *clean* edges of H, i.e., edges that do not appear in any triangle of G. Note that all but at most $2 \cdot |D|$ edges of H are clean, because every edge of H that is not incident to any edge in D is necessarily clean. Crucially, endpoints of a clean edge can only occur together in clauses of size at most 2. This property allows us to simplify formula Q significantly (using three reduction rules, whose correctness follows by using Hintikka strategies [55]) s.t. we can assume the endpoints of every clean edge are contained in exactly one clause of Q. This allows us to introduce a simple reduction rule for removing every leaf of H that is not an endpoint of an edge in D. Then, in the reduced instance, H has at most $2{\cdot}|D|$ leaves and therefore at most $2{\cdot}|D|{-}2$ vertices of *degree* (number of adjacent vertices) larger than 2. Our last reduction rule, which is involved and based on Hintikka strategies, allows us to reduce degree 2 vertices in H by showing that any maximal (clean) path of degree 2 vertices in H must contain a variable (the innermost variable), that can be "removed". This shows that the size of H and therefore also G is bounded, which in turn allows us to obtain a bound for Q.

From Theorem V.2, we know that we can brute-force on the kernel (output) after preprocessing. Immediately, we obtain a single-exponential fpt algorithm for CQSAT.

Corollary V.3. CQSAT *is fpt parameterized by the feedback edge number of the primal graph.*

Interestingly, our kernelization even provides a linear kernel, i.e., the size of the kernel depends only linearly on the parameter, if we restrict ourselves to c-CQBFs.

Corollary V.4. *Let c be an integer; Q be a c-*CQBF*. In polynomial time, we obtain an equivalent c-*CQBF *with at most $12k-8$ variables and $10k-9+3k\ (3^c/\binom{c}{2})$ clauses, s.t. k is the size of a smallest FES of G_Q.*

Similarly, we can show the existance of a smaller kernel for the feedback edge number of the incidence graph.

Theorem V.5 ($\star$)**.** *Let Q be a* CQBF *with feedback edge number k of I_Q. In polynomial time, we can construct an equivalent* CQBF *with $\leq 24k-17$ variables and clauses.*

VI. TRACTABILTY FOR CQSAT ON PRIMAL GRAPHS

Above, our results draw a comprehensive picture of the fine-grained complexity of CQSAT with respect to the incidence graph. However, when considering the primal graph there is a gap between the tractability for vertex cover number and feedback edge number and the known intractability for treewidth. To address this, one may ask what is the complexity of CQSAT with respect to parameters feedback vertex number and treedepth? We progress towards resolving the question for treedepth, which is not only completely open, but existing techniques do not even allow us to solve the problem for significant restrictions of treedepth. Such a parameter is c-deletion set, i.e., deletion distance to components of size at most c, which is well-known to be inbetween vertex cover number and treedepth, see related work on *vertex integrity* [56]. We provide three novel algorithms, each representing a step towards generalizing the tractability of CQSAT for vertex cover number. Each uses a different approach providing new insights that are promising for treedepth.

First, we show that different variants of c-deletion sets can be efficiently computed, which we achieve by the following proposition. Let $\mathcal{P}(Q, D, c)$ be any property that can be true or false for a CQBF Q and c-deletion set D of Q. We say $\mathcal{P}$ is *efficiently computable* if there is an algorithm that given Q, D, and c decides whether $\mathcal{P}(Q, D, c)$ holds in fpt-time parameterized by $|D| + c$.

Proposition VI.1 ($\star$)**.** *Let $\mathcal{P}$ be any efficiently computable property and let Q be a* CQBF*. Then, computing a smallest c-deletion set D of Q that satisfies $\mathcal{P}(Q, D, c)$ is fixed-parameter tractable parameterized by $|D| + c$.*

We can eliminate all universal variables in a c-deletion set D of a CQBF without losing the structure of the formula, i.e., we obtain a formula, which is not too large and still has a $2^c c$-deletion set of size at most $2^c c$.

Proposition VI.2 ($\star$)**.** *Let Q be a* CQBF *and let D be a c-deletion set for Q. Then, in time $\mathcal{O}(2^u \|Q\|)$, where $u = |D \cap \mathrm{var}^\forall(Q)|$, we can construct an equivalent* CQBF *Q' and a set $D' \subseteq \mathrm{var}^\exists(Q')$ with $|D'| \leq 2^u |D|$ s.t. D' is a $2^u c$-deletion set for Q'.*

A. Components of Type $\exists^{\leq 1}\forall$

CQSAT is fpt parameterized by the size of a c-deletion set into components of the form $\exists^{\leq 1}\forall$, i.e., components have at most one existential variable occurring before all its (arbitrarily many) universal variables in the prefix.

Theorem VI.3. CQSAT *is fixed-parameter tractable parameterized by $k + c$, where k is the size of a smallest c-deletion set into components of the form $\exists^{\leq 1}\forall$.*

This generalizes fixed-parameter tractability of CQSAT parameterized by vertex cover, since every component can have arbitrary many variables as well as one quantifier alternation, as opposed to containing only one variable.

Checking whether every component is of the form $\exists^{\leq 1}\forall$ can be achieved in polynomial time. Since we can compute a smallest c-deletion set into components of the form $\exists^{\leq 1}\forall$ in fpt-time parameterized by its size plus c due to Proposition VI.1, it suffices to show the following.

Theorem VI.4 ($\star$)**.** *Let Q be a* CQBF *and $D \subseteq \mathrm{var}(Q)$ be a c-deletion set for Q into components of the form $\exists^{\leq 1}\forall$. Then, deciding Q is fpt parameterized by $|D| + c$.*

The main ingredient for the proof of Theorem VI.4 is Lemma XIII.3 ($\star$). It allows us to remove all but at most 2^c components of every component type. Together with bounding the number of component types, we reduce Q to a bounded-size formula that we brute-force.

B. Single-Variable Deletion Sets

Next, we consider deletion sets that consist only of a single variable e, but where the quantifier prefix restricted to variables occurring in a component can have an arbitrary shape. Without loss of generality we assume e is existentially quantified and innermost (quantifier prefix).

We show this by an evaluation game where a universal player and an existential player take turns assigning their respective variables, in the order of the quantifier prefix. The universal player seeks to assign such that no assignment of e is left for the existential player to satisfy all clauses. If universal has a strategy ensuring some assignment of e cannot be played by existential, we say the strategy *forbids* this assignment. The QBF is false if and only if universal can forbid both assignments.

Since components do not share universal variables, the universal strategy can be decomposed into strategies played in the individual components. If there are distinct components where universal can forbid assignments $e \mapsto 0$ and $e \mapsto 1$, respectively, corresponding strategies can be composed into a universal winning strategy. An interesting case arises if there is a single component where universal must choose an assignment to forbid, and existential must similarly choose which assignment to play in remaining components. Universal wins if and only if the latest point, that is the innermost variable, where they can forbid, comes after the latest point where existential can choose (play). A formal development of these intuitions leads to the following (see Appendix XIII-B).

Theorem VI.5 ($\star$). *Let Q be a* CQBF *with a c-deletion set of size* 1. *Deciding Q is fpt parameterized by c.*

C. Formulas with Many Components of Each Type

Let Q be a CQBF and let $D \subseteq \mathrm{var}(Q)$ be a c-deletion set for Q. Moreover, let Q' and D' be the CQBF and $2^c c$-deletion set of Q' of size at most $2^c|D|$ obtained after eliminating all universal variables in D using Proposition VI.2. We say D' is *universally complete* if every component type of Q' consists of at least $2^{|D'|}$ components. We show CQSAT to be fpt parameterized by the size of a smallest universally complete c-deletion set.

Theorem VI.6. CQSAT *is fpt by $k + c$, where k is the size of a smallest universally complete c-deletion set.*

This result is surprising and counter-intuitive at first, as it seems to indicate that deciding a large CQBF Q (with many components of each type) is simple, while we do not know whether this holds for sub-formulas of Q. However, we show that many components of the same type allows the universal player to play all possible local counter-strategies for each type. This makes the relative ordering of various components in the prefix irrelevant.

Note that deciding whether a given c-deletion set is universally complete is an efficiently computable property due to Proposition VI.2. This implies that we can compute a smallest universally complete c-deletion set in fpt-time parameterized by its size plus c by Proposition VI.1. So, to show Theorem VI.6, we assume we are given a smallest universally complete c-deletion set D for Q. Moreover, by Proposition VI.2, we can assume we are given the corresponding Q' and D'. It suffices to show:

Theorem VI.7 ($\star$). *Let Q be a* CQBF *and let $D \subseteq \mathrm{var}^{\exists}(Q)$ be a universally complete c-deletion set for Q. Then deciding Q is fpt parameterized by $|D| + c$.*

VII. Conclusion

We consider evaluating quantified Boolean formulas (QSAT) under structural restrictions. While the classical complexity and the parameters treewidth and vertex cover number are well understood on primal graphs, we address the incidence graph and the gap between both parameters. We provide new upper and lower bounds and establish a comprehensive complexity-theoretic picture for QSAT concerning the most fundamental graph-structural parameters of this graph. We thereby sharpen the boundaries between parameters where one can drop the quantifier depth in the parameterization and those where one cannot, providing a nearly-complete picture of parameters of the incidence graph, cf., Figure 1. We show lower bounds for feedback vertex number and treedepth by designing structure-aware (SAW) reductions. We then complement known upper bounds for vertex cover number by tractability (fpt) results for feedback edge number.

Despite this paper closing many gaps and providing deeper insights into the hardness of QSAT for structural parameters, it does not fully settle QSAT for parameters of the primal graph. A single clause makes the difference: if we omit edges induced by one clause, our lower bounds for feedback vertex number and treedepth carry over, as indicated in Figure 2 (left). As a first step towards filling this gap and analyzing treedepth of the primal graph, we establish fpt for variants of the deletion set parameter.

Techniques: While the ideas behind structure-aware (SAW) reductions have been implicitly used in limited contexts, e.g., [26], we formalize and fully develop the technique. We establish a template to design specific self-reductions from QSAT to QSAT when we are interested in precise lower bounds under the exponential time hypothesis (ETH) for various parameters. As illustrated in Figure 2 (right), SAW reductions allow us to trade an exponential decrease (log in the exponent) of one parameter for an exponential increase (increasing the tower height) of runtime dependency on a second parameter.

Future Work: Our analysis opens up several interesting questions: Is QSAT on CNFs fpt parameterized by either the feedback vertex number or the treedepth of the primal graph? We have indications for both possible outcomes. (N) Our lower bounds are close since allowing merely one additional clause yields intractability for treedepth and feedback vertex number. This points toward hardness, and our SAW-reductions might provide a good starting point to understand and obtain intractability for one or both parameters on the primal graph. (Y) The algorithmic techniques we developed for variants of the c-deletion set, point in the other direction may serve as the underpinning for an fpt result. While our lower bounds for feedback vertex number of the incidence graph are tight under ETH, this is open for treedepth. Besides, lower bounds under SETH might be interesting. A further question is whether our techniques carry over to DQBF [57] and QCSP. Finally, we expect SAW reductions to be a useful tool for problems within the polynomial hierarchy. Indeed, many problems of practical interest would benefit from precise bounds; see, e.g., [16], [26]. Some are highly relevant in other communities, e.g., explainability [58].

References

[1] M. Benedetti and H. Mangassarian, "QBF-based formal verification: Experience and perspectives," *J. Satisf. Boolean Model. Comput.*, vol. 5, no. 1-4, pp. 133–191, 2008. [Online]. Available: https://doi.org/10.3233/sat190055

[2] R. Bloem, U. Egly, P. Klampfl, R. Könighofer, and F. Lonsing, "Sat-based methods for circuit synthesis," in *Formal Methods in Computer-Aided Design, FMCAD 2014, Lausanne, Switzerland, October 21-24, 2014*. IEEE, 2014, pp. 31–34. [Online]. Available: https://doi.org/10.1109/FMCAD.2014.6987592

[3] U. Egly, T. Eiter, H. Tompits, and S. Woltran, "Solving advanced reasoning tasks using quantified boolean formulas," in *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA.*, 2000, pp. 417–422.

[4] C. Otwell, A. Remshagen, and K. Truemper, "An effective QBF solver for planning problems," in *Proceedings of the International Conference on Modeling, Simulation & Visualization Methods, MSV '04 & Proceedings of the International Conference on Algorithmic Mathematics & Computer Science, AMCS '04, June 21-24, 2004, Las Vegas, Nevada, USA*. CSREA Press, 2004, pp. 311–316.

[5] J. Rintanen, "Constructing conditional plans by a theorem-prover," *J. Artif. Intell. Res.*, vol. 10, pp. 323–352, 1999.

[6] A. Shukla, A. Biere, L. Pulina, and M. Seidl, "A survey on applications of quantified boolean formulas," in *31st IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2019*. IEEE, 2019, pp. 78–84. [Online]. Available: https://doi.org/10.1109/ICTAI.2019.00020

[7] I. P. Gent, P. Nightingale, A. Rowley, and K. Stergiou, "Solving quantified constraint satisfaction problems," *Artif. Intell.*, vol. 172, no. 6, pp. 738–771, 2008.

[8] R. Dechter, "Tractable structures for constraint satisfaction problems," in *Handbook of Constraint Programming*. Elsevier, 2006, vol. I, ch. 7, pp. 209–244.

[9] E. C. Freuder, "A sufficient condition for backtrack-bounded search," *J. ACM*, vol. 32, no. 4, pp. 755–761, 1985.

[10] M. Grohe, "The parameterized complexity of database queries," in *Proceedings of the 29th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, ser. PODS '01. ACM, 2001, pp. 82–92.

[11] G. Tseytin, "On the complexity of derivation in propositional calculus," *Studies in Constructive Mathematics and Mathematical Logic, Part II, Seminars in Mathematics*, pp. 115–125, 1970, translated from Russian: Zapiski Nauchnykh Seminarov LOMI 8 (1968), pp. 234–259.

[12] V. V. Vazirani, *Approximation Algorithms*. Springer, 2003.

[13] S. Toda, "PP is as hard as the polynomial-time hierarchy," *SIAM J. Comput.*, vol. 20, no. 5, pp. 865–877, 1991.

[14] C. H. Papadimitriou, *Computational Complexity*. Addison-Wesley, 1994.

[15] C. Lautemann, "BPP and the polynomial hierarchy," *Information Processing Letters*, vol. 17, no. 4, pp. 215–217, 1983. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0020019083900443

[16] G. Pan and M. Y. Vardi, "Fixed-parameter hierarchies inside PSPACE," in *LICS*. IEEE Computer Society, 2006, pp. 27–36.

[17] A. Atserias and S. Oliva, "Bounded-width QBF is PSPACE-complete," *J. Comput. Syst. Sci.*, vol. 80, no. 7, pp. 1415–1429, 2014.

[18] M. Samer and S. Szeider, "Algorithms for propositional model counting," *J. Discrete Algorithms*, vol. 8, no. 1, pp. 50–64, 2010.

[19] ——, "Fixed-parameter tractability," in *Handbook of Satisfiability - Second Edition*, ser. Frontiers in Artificial Intelligence and Applications, A. Biere, M. Heule, H. van Maaren, and T. Walsh, Eds. IOS Press, 2021, vol. 336, pp. 693–736. [Online]. Available: https://doi.org/10.3233/FAIA201000

[20] R. Wallon and S. Mengel, "Revisiting graph width measures for CNF-encodings," *J. Artif. Intell. Res.*, vol. 67, pp. 409–436, 2020. [Online]. Available: https://doi.org/10.1613/jair.1.11750

[21] R. Ganian and S. Ordyniak, "The complexity landscape of decompositional parameters for ILP," *Artif. Intell.*, vol. 257, pp. 61–71, 2018. [Online]. Available: https://doi.org/10.1016/j.artint.2017.12.006

[22] ——, "Solving integer linear programs by exploiting variable-constraint interactions: A survey," *Algorithms*, vol. 12, no. 12, p. 248, 2019. [Online]. Available: https://doi.org/10.3390/a12120248

[23] T. F. N. Chan, J. W. Cooper, M. Koutecký, D. Král, and K. Pekárková, "Matrices of optimal tree-depth and a row-invariant parameterized algorithm for integer programming," *SIAM J. Comput.*, vol. 51, no. 3, pp. 664–700, 2022. [Online]. Available: https://doi.org/10.1137/20m1353502

[24] R. Dechter and J. Pearl, "Tree clustering for constraint networks," *Artif. Intell.*, vol. 38, no. 3, pp. 353–366, 1989. [Online]. Available: https://doi.org/10.1016/0004-3702(89)90037-4

[25] M. Samer and S. Szeider, "Constraint satisfaction with bounded treewidth revisited," *J. Comput. Syst. Sci.*, vol. 76, no. 2, pp. 103–114, 2010.

[26] J. K. Fichte, M. Hecher, and A. Pfandler, "Lower bounds for qbfs of bounded treewidth," in *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science*. ACM, 2020, pp. 410–424. [Online]. Available: https://doi.org/10.1145/3373718.3394756

[27] M. Lampis and V. Mitsou, "Treewidth with a quantifier alternation revisited," in *Proceedings of the 12th International Symposium on Parameterized and Exact Computation (IPEC'17)*, ser. LIPIcs, vol. 89. Dagstuhl Publishing, 2017, pp. 26:1–26:12.

[28] J. Nešetřil and P. O. de Mendez, *Sparsity - Graphs, Structures, and Algorithms*, ser. Algorithms and combinatorics. Springer, 2012, vol. 28.

[29] F. Slivovsky and S. Szeider, "A faster algorithm for propositional model counting parameterized by incidence treewidth," in *23rd International Conference on Theory and Applications of Satisfiability Testing (SAT 2020)*, ser. Lecture Notes in Computer Science, vol. 12178. Springer, 2020, pp. 267–276. [Online]. Available: https://doi.org/10.1007/978-3-030-51825-7_19

[30] R. de Haan, I. A. Kanj, and S. Szeider, "On the subexponential-time complexity of CSP," *J. Artif. Intell. Res.*, vol. 52, pp. 203–234, 2015. [Online]. Available: https://doi.org/10.1613/jair.4540

[31] E. Eiben, R. Ganian, D. Knop, S. Ordyniak, M. Pilipczuk, and M. Wrochna, "Integer programming and incidence treedepth," in *Integer Programming and Combinatorial Optimization - 20th International Conference, IPCO 2019, Ann Arbor, MI, USA, May 22-24, 2019, Proceedings*, ser. Lecture Notes in Computer Science, A. Lodi and V. Nagarajan, Eds., vol. 11480. Springer, 2019, pp. 194–204. [Online]. Available: https://doi.org/10.1007/978-3-030-17953-3_15

[32] F. Capelli and S. Mengel, "Tractable QBF by knowledge compilation," in *STACS 2019*, ser. LIPIcs, R. Niedermeier and C. Paul, Eds., vol. 126. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019, pp. 18:1–18:16. [Online]. Available: https://doi.org/10.4230/LIPIcs.STACS.2019.18

[33] R. Impagliazzo, R. Paturi, and F. Zane, "Which problems have strongly exponential complexity?" *J. Comput. Syst. Sci.*, vol. 63, no. 4, pp. 512–530, 2001.

[34] M. Sorge and M. Weller, "The graph parameter hierarchy," 2012–2020. [Online]. Available: https://manyu.pro/assets/parameter-hierarchy.pdf

[35] H. de Rider, "Information System on Graph Classes and their Inclusions (ISGCI)," 2001–2014. [Online]. Available: https://graphclasses.org

[36] C. Ansótegui, C. P. Gomes, and B. Selman, "The achilles' heel of QBF," in *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, M. M. Veloso and S. Kambhampati, Eds. AAAI Press / The MIT Press, 2005, pp. 275–281. [Online]. Available: http://www.aaai.org/Library/AAAI/2005/aaai05-044.php

[37] C. Jordan, W. Klieber, and M. Seidl, "Non-CNF QBF solving with QCIR," in *Proceedings of the AAAI Workshops at the 13th*

AAAI Conference on Artificial Intelligence (AAAI'16). The AAAI Press, 2016, pp. 320–326.
[38] A. Goultiaeva, M. Seidl, and A. Biere, "Bridging the gap between dual propagation and cnf-based QBF solving," in *Design, Automation and Test in Europe, DATE 13, Grenoble, France, March 18-22, 2013*, E. Macii, Ed. EDA Consortium San Jose, CA, USA / ACM DL, 2013, pp. 811–814. [Online]. Available: https://doi.org/10.7873/DATE.2013.172
[39] L. J. Stockmeyer and A. R. Meyer, "Word problems requiring exponential time," in *Proceedings of the 5th Annual ACM Symposium on Theory of Computing (STOC'73)*. ACM, 1973, pp. 1–9.
[40] H. Chen, "Quantified constraint satisfaction and bounded treewidth," in *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI'04)*, vol. IOS Press, 2004, pp. 161–170.
[41] R. Diestel, *Graph Theory, 4th Edition*, ser. Graduate Texts in Mathematics. Springer, 2012, vol. 173.
[42] J. A. Bondy and U. S. R. Murty, *Graph theory*, ser. Graduate Texts in Mathematics. Springer, 2008, vol. 244.
[43] J. Flum and M. Grohe, *Parameterized Complexity Theory*, ser. Theor. Comput. Sci. Springer, 2006.
[44] R. Niedermeier, *Invitation to Fixed-Parameter Algorithms*, ser. Oxford Lecture Series in Mathematics and its Applications. New York, NY, USA: Oxford University Press, 2006, vol. 31.
[45] A. Biere, M. Heule, H. van Maaren, and T. Walsh, Eds., *Handbook of Satisfiability*, ser. FAIA. IOS Press, 2009, vol. 185.
[46] H. Kleine Büning and T. Lettman, *Propositional Logic: Deduction and Algorithms*. New York, NY, USA: Cambridge University Press, 1999.
[47] N. Robertson and P. D. Seymour, "Graph minors. I. Excluding a forest," *J. Comb. Theory, Ser. B*, vol. 35, no. 1, pp. 39–61, 1983.
[48] ——, "Graph minors. X. Obstructions to tree-decomposition," *J. Comb. Theory, Ser. B*, vol. 52, no. 2, pp. 153–190, 1991.
[49] P. Beame, R. Impagliazzo, T. Pitassi, and N. Segerlind, "Formula caching in DPLL," *ACM Trans. Comput. Theory*, vol. 1, no. 3, pp. 9:1–9:33, 2010. [Online]. Available: https://doi.org/10.1145/1714450.1714452
[50] M. M. Syslo, "Characterizations of outerplanar graphs," *Discret. Math.*, vol. 26, no. 1, pp. 47–53, 1979. [Online]. Available: https://doi.org/10.1016/0012-365X(79)90060-8
[51] D. Marx and V. Mitsou, "Double-Exponential and Triple-Exponential Bounds for Choosability Problems Parameterized by Treewidth," in *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, ser. LIPIcs, vol. 55. Dagstuhl Publishing, 2016, pp. 28:1–28:15.
[52] M. Pilipczuk and M. Sorge, "A double exponential lower bound for the distinct vectors problem," *Discret. Math. Theor. Comput. Sci.*, vol. 22, no. 4, 2020. [Online]. Available: http://dmtcs.episciences.org/6789
[53] B. Aspvall, M. F. Plass, and R. E. Tarjan, "A linear-time algorithm for testing the truth of certain quantified Boolean formulas," *Inf. Process. Lett.*, vol. 8, no. 3, pp. 121–123, 1979.
[54] R. Williams, "Algorithms for quantified Boolean formulas," in *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 6-8, 2002, San Francisco, CA, USA*, D. Eppstein, Ed. ACM/SIAM, 2002, pp. 299–307.
[55] E. Grädel, P. G. Kolaitis, L. Libkin, M. Marx, J. Spencer, M. Y. Vardi, Y. Venema, and S. Weinstein, *Finite Model Theory and Its Applications*. Springer, 2005.
[56] M. Lampis and V. Mitsou, "Fine-grained meta-theorems for vertex integrity," in *32nd International Symposium on Algorithms and Computation, ISAAC 2021, December 6-8, 2021, Fukuoka, Japan*, ser. LIPIcs, vol. 212. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, pp. 34:1–34:15.
[57] S. Azhar, G. Peterson, and J. Reif, "Lower bounds for multiplayer non-cooperative games of incomplete information," *Journal of Computers and Mathematics with Applications*, vol. 41, pp. 957 – 992, 2001.
[58] A. Darwiche, "Three modern roles for logic in AI," in *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS'20)*. New York, NY, USA: ACM, 2020, pp. 229–243.
[59] R. G. Downey and M. R. Fellows, *Fundamentals of Parameterized Complexity*, ser. Texts in Computer Science. Springer, 2013.
[60] T. Korhonen, "Single-Exponential Time 2-Approximation Algorithm for Treewidth," in *Proceedings of the 22nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, vol. abs/2104.07463, 2022.
[61] J. Li and J. Nederlof, "Detecting feedback vertex sets of size k in $O^*(2.7^k)$ time," in *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, (SODA 2020)*, S. Chawla, Ed. SIAM, 2020, pp. 971–989. [Online]. Available: https://doi.org/10.1137/1.9781611975994.58
[62] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Parameterized Algorithms*. Springer, 2015.
[63] F. Reidl, P. Rossmanith, F. S. Villaamil, and S. Sikdar, "A faster parameterized algorithm for treedepth," in *In Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP 2014)*, ser. Lecture Notes in Computer Science, vol. 8572. Springer, 2014, pp. 931–942.
[64] P. G. Drange, M. S. Dregi, and P. van 't Hof, "On the computational complexity of vertex integrity and component order connectivity," *Algorithmica*, vol. 76, no. 4, pp. 1181–1202, 2016.