QCDCL with Cube Learning or Pure Literal Elimination – What is Best?

Benjamin Böhm[1], **Tomáš Peitl**[2], **Olaf Beyersdorff**[1]
[1]Friedrich Schiller University Jena, Germany
[2]TU Wien, Vienna, Austria
benjamin.boehm@uni-jena.de, peitl@ac.tuwien.ac.at, olaf.beyersdorff@uni-jena.de

Abstract

Quantified conflict-driven clause learning (QCDCL) is one of the main approaches for solving quantified Boolean formulas (QBF). We formalise and investigate several versions of QCDCL that include cube learning and/or pure-literal elimination, and formally compare the resulting solving models via proof complexity techniques. Our results show that almost all of the QCDCL models are exponentially incomparable with respect to proof size (and hence solver running time), pointing towards different orthogonal ways how to practically implement QCDCL.

1 Introduction

SAT solving has revolutionised the way we perceive computationally hard problems. Determining the satisfiability of propositional formulas (SAT) has traditionally been viewed as intractable due to its NP completeness. In contrast, modern SAT solvers today routinely solve huge industrial instances of SAT from a wide variety of application domains [Biere *et al.*, 2021a]. This success of solving has not stopped at SAT, but in the last two decades was lifted to increasingly more challenging computational settings, with solving quantified Boolean formulas (QBF)—a PSPACE-complete problem—receiving key attention [Beyersdorff *et al.*, 2021].

Conflict driven clause learning (CDCL) is the main paradigm of modern SAT solving [Marques Silva *et al.*, 2021]. Based on the classic DPLL algorithm from the 1960s, it combines a number of advanced features, including clause learning, efficient Boolean constraint propagation, decision heuristics, restart strategies, and many more. In QBF there exist several competing approaches to solving, with lifting CDCL to the quantified level in the form of QCDCL as one of the main paradigms [Zhang and Malik, 2002], implemented e.g. in the state-of-the-art solvers DepQBF [Lonsing and Egly, 2017] and Qute [Peitl *et al.*, 2019].

For SAT/QBF solving, two questions of prime theoretical and practical importance are: (1) why are SAT/QBF solvers so effective and on which formulas do they fail? (2) Which solving ingredients are most important for their performance?

For (1), proof complexity offers the main theoretical approach to analyse the strength of solving [Buss and Nordström, 2021]. In a breakthrough result, Pipatsrisawat and Darwiche [2011] and Atserias *et al.* [2011] established that CDCL on unsatisfiable formulas is equivalent to the resolution proof system, in the sense that from a CDCL run a resolution proof can be efficiently extracted [Beame *et al.*, 2004], and conversely, each resolution proof can be efficiently simulated by CDCL [Pipatsrisawat and Darwiche, 2011]. Hence the well-developed proof-complexity machinery for proof size lower bounds in resolution [Krajíček, 2019] is directly applicable to show lower bounds for running time in CDCL.

The latter simulation of resolution by CDCL assumes a strong ‘non-deterministic’ version of CDCL, whereas practical CDCL (using decision heuristics such as VSIDS) has been recently proved to be exponentially weaker than resolution [Vinyals, 2020]. In contrast, an analogous proof-theoretic characterisation is not known for QCDCL, and in particular QCDCL has recently been shown to be incomparable to Q-Resolution [Beyersdorff and Böhm, 2021], the QBF analogue of propositional resolution [Kleine Büning *et al.*, 1995].

Regarding question (2) above, there are some experimental studies [Sakallah and Marques-Silva, 2011; Elffers *et al.*, 2018; Kokkala and Nordström, 2020], but no rigorous theoretical results are known on which (Q)CDCL ingredients are most crucial for performance. Of course, gaining such a theoretical understanding would also be very valuable in guiding future solving developments.

In this paper, we contribute towards question (2) in QBF.

1.1 Contributions

Following the approach of Beyersdorff and Böhm [2021], we model QCDCL as rigorously defined proof systems that are amenable to a proof-complexity analysis. This involves formalising individual QCDCL ingredients, such as clause and cube learning and different variants of Boolean constraint propagation. These can then be ‘switched’ on or off, resulting in a number of different QCDCL solving models that we can formally investigate. Our results can be summarised as follows.

(a) QCDCL with or without cube learning. In contrast to SAT solving, where there is somewhat of an asymmetry between satisfiable and unsatisfiable formulas, QCDCL implements a dual approach for false and true QBFs. In addition to learning clauses (as in CDCL) when running into a conflict under the current assignment, QCDCL also learns terms (or cubes) in the case a satisfying assignment is found (or a previously learned cube is satisfied). While cube learning is necessary to make QCDCL solving complete on true QBFs, it is less clear what the effect of cube learning is on false QBFs (and we only consider those throughout the paper as we cast all our models in terms of refutational proof systems, in accordance with the proof complexity analysis of SAT [Buss and Nordström, 2021]). Here we establish the perhaps surprising result that even for false QBFs, cube learning can be advantageous, in the sense that QCDCL without cube learning (as a proof system for false QBFs) is exponentially weaker than QCDCL with cube learning.

(b) QCDCL with or without pure-literal elimination. In its simplest form, Boolean constraint propagation, used to construct trails in (Q)CDCL, implements unit propagation. However, further methods can be additionally employed (and are considered in pre- and in-processing [Biere *et al.*, 2021b]). One of the classic mechanisms is pure-literal elimination, setting a pure literal (which occurs in only one polarity) to the obvious value. This is e.g. implemented in DepQBF and an efficient implementation is described by Lonsing [2012].

We show that QCDCL with or without pure-literal elimination results in incomparable proof systems, i.e., there are QBFs that are easy in QCDCL with pure literal elimination, but hard in plain QCDCL, and vice versa (the latter is perhaps more surprising).

(c) Comparing QCDCL extensions. Given the preceding results, it is natural (and possibly most interesting for practice) to ask how the different QCDCL extensions compare with each other. We consider QCDCL with cube learning, QCDCL with pure-literal elimination but without cube learning, and QCDCL with both cube learning and pure-literal elimination. Except for the simulation of the second by the third system, we again obtain incomparability results between the systems with exponential separations. We further show that all these systems are incomparable to Q-Resolution, again via exponential separations. An overview of the systems and their relations is given in Figure 1.

Technically, our results rest on formalising QCDCL systems as proof calculi and exhibiting specific QBFs for their separations. The latter includes both the explicit construction of short QCDCL runs and proving exponential proof size lower bounds for the relevant calculi. For the lower bounds, we identify a property of proofs (called primitivity here) that allows to use proof-theoretic machinery of Böhm and Beyersdorff [2021] in the context of our QCDCL systems.

Our theoretical results on the strength of different QCDCL models are empirically confirmed by experiments with state-of-the-art QCDCL solvers (cf. Section 8).

Organisation. We start in Section 2 by reviewing QBFs and Q-Resolution. In Section 3 we model variants of QCDCL as formal proof systems and develop a lower technique for such systems in Section 4. Sections 5 to 8 then contain our results on the relative strength of QCDCL variants. We conclude in Section 9 with an outlook on future research.

Due to space constraints, some details are omitted.

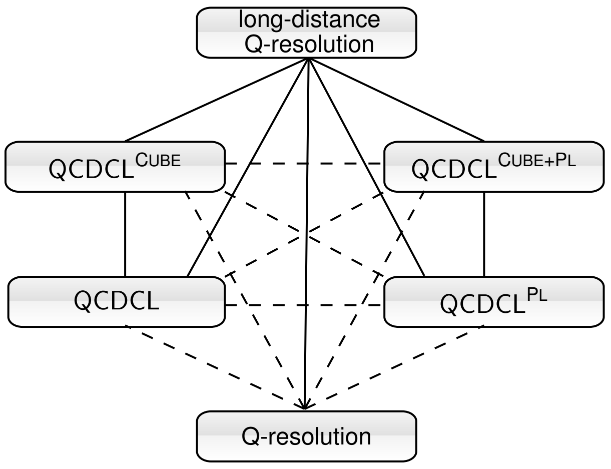

Figure 1: Hasse diagram of the simulation order of QCDCL proof systems. Solid lines represent p-simulations and exponential separations (where the system depicted above is the stronger one). Dashed lines represent separations in both directions (i.e., incomparability).

2 Preliminaries

We will use standard notions from propositional logic, such as *variables*, *literals*, *(propositional) formulas*, *clauses*, *conjunctive normal form* (*CNF*), *disjunctive normal form* (*DNF*), *assignments*, *satisfiability* or *restrictions*. A *cube* (or *term*) is a conjunction of literals. A literal ℓ in a formula Φ is *pure*, if it only appears in one polarity (i.e., if ℓ is contained in Φ, but $\bar{\ell}$ is not). We define two "empty" literals $\bot$ and $\top$.

A *QBF* (quantified Boolean formula) $\Phi = \mathcal{Q} \cdot \phi$ consists of a propositional formula ϕ, called the *matrix* (denoted by $\mathfrak{C}(\Phi)$), and a *prefix* $\mathcal{Q}$. A *prefix* $\mathcal{Q} = \mathcal{Q}'_1 V_1 \ldots \mathcal{Q}'_s V_s$ consists of non-empty and pairwise disjoint sets of variables $V_1, \ldots, V_s$ and quantifiers $\mathcal{Q}'_1, \ldots, \mathcal{Q}'_s \in \{\exists, \forall\}$ with $\mathcal{Q}'_i \neq \mathcal{Q}'_{i+1}$ for $i \in [s-1]$. For a variable x in $\mathcal{Q}$, the *quantifier level* is $\mathrm{lv}(x) := \mathrm{lv}_\Phi(x) := i$, if $x \in V_i$. For $\mathrm{lv}_\Phi(\ell_1) < \mathrm{lv}_\Phi(\ell_2)$ we write $\ell_1 <_\Phi \ell_2$.

We use the proof systems Q-resolution [Kleine Büning *et al.*, 1995] and long-distance Q-resolution [Balabanov and Jiang, 2012], containing resolution and reduction rules. In general, a clause C can be reduced universally, while a cube D can be reduced existentially. We denote the maximally universally (resp. existentially) reduced clause (resp. cube) by $\mathrm{red}^{\forall}_{\Phi}(C)$ (resp. $\mathrm{red}^{\exists}_{\Phi}(D)$).

3 Formal Calculi for QCDCL Versions

In this section we model different versions of QCDCL as formal proof systems (we sketch this only here; for background on QCDCL cf. [Beyersdorff *et al.*, 2021]). For this we need to formalise QCDCL ingredients. We start with trails. A *trail* $\mathcal{T}$ for a QCNF Φ is a finite sequence of literals from Φ, including the empty literals $\bot$ and $\top$. In general, a trail has the form

$$\begin{aligned}\mathcal{T} = (&p_{(0,1)}, \ldots, p_{(0,g_0)}; \mathbf{d_1}, p_{(1,1)}, \ldots, \\ &p_{(1,g_1)}; \ldots; \mathbf{d_r}, p_{(r,1)}, \ldots, p_{(r,g_r)}),\end{aligned} \tag{3.1}$$

where the d_i are *decision literals* and $p_{(i,j)}$ are *propagated literals*. Decision literals are written in **boldface**. We use a semicolon before each decision to mark the end of a decision level. We write $x <_{\mathcal{T}} y$ if $x, y \in \mathcal{T}$ and x is left of y in $\mathcal{T}$.

Trails can be interpreted as non-tautological sets of literals, and therefore as (partial) assignments. If $\mathcal{T}$ is a trail, then $\mathcal{T}[i, j]$, for $i \in \{0, \dots, r\}$ and $j \in \{0, \dots, g_i\}$, is defined as the *subtrail* that contains all literals from $\mathcal{T}$ left of (and excluding) $p_{(i,j)}$ (resp. d_i, if $j = 0$). A trail $\mathcal{T}$ has *run into conflict* if $\bot \in \mathcal{T}$ or $\top \in \mathcal{T}$.

Simply put, our QCDCL proof systems can be interpreted as sequences of trails. These trails cannot be created arbitrary, but have to follow special rules, depending on the model. We consider the following four QCDCL variants:

- QCDCL, which can be seen as the plain model where we can only make decisions following the level order of the quantifier prefix, make propagations using clauses and use classic clause learning. We will never learn or use cubes and pure-literal elimination is turned off.
- QCDCL$^{\mathsf{CUBE}}$ is an extension of QCDCL in which we can learn cubes and use them for propagations. Decisions are still level-ordered and pure-literal elimination is turned off.
- QCDCL$^{\mathsf{PL}}$ is an extension of QCDCL, where we decide literals out of order if they are pure in the current configuration (pure-literal elimination). All other decisions (which we call regular decisions) are still level ordered. Cube learning is turned off.
- QCDCL$^{\mathsf{CUBE+PL}}$ is an extension of QCDCL$^{\mathsf{PL}}$, in which cube learning is now allowed (as in QCDCL$^{\mathsf{CUBE}}$).

Note that decisions can only be made if there are no more propagations possible and pure literal decisions always have a higher priority than regular decisions. Also, conflicts have a higher priority than propagations of proper (existential or universal) literals. Hence, we will never skip conflicts, propagations or pure literal decisions. For each propagated literal $p_{(i,j)}$ in a trail $\mathcal{T}$ the formula must contain a clause or a cube that caused this propagation by becoming a unit clause/cube. We denote such a clause/cube by $\text{ante}_{\mathcal{T}}(p_{(i,j)})$.

After a trail has run into a conflict, or if all variables were assigned, we can start the learning process.

Definition 1 (learnable constraints)**.** *Let $\mathcal{T}$ be a trail for Φ of the form* (3.1) *with $p_{(r,g_r)} \in \{\bot, \top\}$. Starting with ante$_{\mathcal{T}}(\bot)$ (resp. ante$_{\mathcal{T}}(\top)$) we reversely resolve over the antecedent clauses (cubes) that propagated the existential (universal) variables, until we stop at some arbitrarily chosen point. The clause (cube) we so derive is a* learnable constraint*. We denote the set of learnable constraints by $\mathfrak{L}(\mathcal{T})$.*

We can also learn cubes from trails that did not run into conflict. If $\mathcal{T}$ is a total assignment of the variables from Φ, then we define the set of learnable constraints as the set of cubes $\mathfrak{L}(\mathcal{T}) := \{red_{\Phi}^{\exists}(D) | \, D \subseteq \mathcal{T}$ and D satisfies $\mathfrak{C}(\Phi)\}$.

Definition 2 (QCDCL proof systems)**.** *Let S be one of* QCDCL, QCDCL$^{\mathsf{CUBE}}$, QCDCL$^{\mathsf{PL}}$, QCDCL$^{\mathsf{CUBE+PL}}$. *An S proof ι from a QCNF $\Phi = \mathcal{Q} \cdot \phi$ of a clause or cube C is a sequence of triples*

$$\iota := [(\mathcal{T}_i, C_i, \pi_i)]_{i=1}^{m},$$

where $C_m = C$, each $\mathcal{T}_i$ is a trail, each $C_i \in \mathfrak{L}(\mathcal{T}_i)$ is one of the constraints we can learn from the trail, and π_i is the long-distance Q-resolution *or* long-distance Q-consensus *proof of C_i we get by performing the steps in Definition 1. We define $\mathfrak{R}(\iota)$ as the proof of C we get by sticking together suitable π_i. We denote the set of trails in ι as $\mathfrak{T}(\iota)$.*

If $C = (\bot)$, then ι is called an S refutation *of Φ. If $C = [\top]$, then ι is an S* verification *of Φ. The proof ends once we have learned $(\bot)$ or $[\top]$. The* size *of ι is $|\iota| := \sum_{i=1}^{m} |\mathcal{T}_i|$.*

Theorem 1. QCDCL, QCDCL$^{\mathsf{CUBE}}$, QCDCL$^{\mathsf{PL}}$, *and* QCDCL$^{\mathsf{CUBE+PL}}$ *are sound and complete proof systems.*

We highlight that these systems formally model QCDCL solving as used in practice (cf. [Beyersdorff *et al.*, 2021]).

4 Proving Lower Bounds for QCDCL Systems

Throughout the paper we will concentrate on Σ_3^b QCNFs which we alway assume to have the form $\Phi = \exists X \forall U \exists T \cdot \phi$ for non-empty blocks of variables X, U, and T.

A literal ℓ is an X-literal, if $\text{var}(\ell) \in X$. Analogously, we get U- and T-literals and variables. A clause $C \in \mathfrak{C}(\Phi)$ is an *X-clause*, if all its literals are X-literals. The empty clause $(\bot)$ is also an X-clause. Analogously, we define *T-clauses*. A clause $C \in \mathfrak{C}(\Phi)$ is an *XT-clause*, if it contains at least one X-literal, at least one T-literal, but no U-literals; analogously we define *UT-clauses*. A clause $C \in \mathfrak{C}(\Phi)$ is an *XUT-clause* if it contains at least one X-, U- and T- literal.

Definition 3. *We say that Φ fulfils the* XT-property*, if $\mathfrak{C}(\Phi)$ contains no XT-clauses, no T-clauses that are unit (or empty) and no two T-clauses from $\mathfrak{C}(\Phi)$ are resolvable.*

As shown by Böhm and Beyersdorff [2021], clause learning does not affect the XT-property, i.e., a formula Φ with the XT-property will still fulfil it during the whole QCDCL run even after having added new clauses to $\mathfrak{C}(\Phi)$.

Next we recall the definition of formula gauge from [Böhm and Beyersdorff, 2021], which represents a measure that can be used for lower bounds.

Definition 4 ([Böhm and Beyersdorff, 2021])**.** *For a QCNF Φ as above let W_Φ be the set of all* Q-resolution *derivations π from Φ of some X-clause such that π only contains resolutions over T-variables and reduction steps. We set gauge(Φ) := $\min\{|C| : C$ is the root of some $\pi \in W_\Phi\}$.*

We now define fully reduced and primitive proofs. Our lower bound technique will then work for fully reduced primitive refutations of formulas that fulfil the XT-property.

Definition 5. *A* long-distance Q-resolution *refutation π of a QCNF Φ is called* fully reduced*, if the following holds: For each clause $C \in \pi$ that contains universal literals that are reducible, the reduction step has to be performed immediately and C cannot be used otherwise in the proof.*

Each proof $\mathfrak{R}(\iota)$ that was extracted from a QCDCL proof ι is automatically fully reduced, as we perform reduction steps as soon as possible during clause learning. On the other hand, primitivity does not hold for proofs $\mathfrak{R}(\iota)$ in general. In fact, the main work in proving our hardness results will be to show that specific extracted proofs are primitive.

Definition 6. *A* long-distance Q-resolution *proof π from a Σ_3^b formula with the XT-property is* primitive*, if there are no two XUT-clauses in π that are resolved over an X-variable.*

Since it is not possible to derive tautological clauses in fully reduced primitive proofs, we may also refer to them as (fully reduced) primitive Q-resolution proofs.

It follows from [Böhm and Beyersdorff, 2021], that these two conditions suffice to show lower bounds via gauge.

Theorem 2 ([Böhm and Beyersdorff, 2021])**.** *Each fully reduced primitive* Q-resolution *refutation of a Σ_3^b QCNF Φ that fulfils the XT-property has size $2^{\Omega(gauge(\Phi))}$.*

The next two results represent the main methodology for most of our hardness results throughout the paper.

Lemma 1. *Let $\mathcal{T}$ be a trail in a* QCDCL, QCDCL^{CUBE}, QCDCL^{PL} *or* QCDCL$^{CUBE+PL}$ *proof from a QCNF Φ with the XT-property. Then for each T-literal $t_1 \in \mathcal{T}$, which was not decided by pure literal elimination, there is a U-literal $u \in \mathcal{T}$ with $u <_{\mathcal{T}} t_1$.*

Proposition 1. *Let ι be a* QCDCL, QCDCL^{CUBE}, QCDCL^{PL} *or* QCDCL$^{CUBE+PL}$ *refutation of a QCNF Φ that fulfils the XT-property. If $\mathfrak{R}(\iota)$ is not primitive, then there exists a trail $\mathcal{T} \in \mathfrak{T}(\iota)$ such that there is a U-literal $u \in \mathcal{T}$ and an X-literal $x \in \mathcal{T}$ with $u <_{\mathcal{T}} x$. Additionally, u cannot be a regular decision literal.*

Basically, this result tells us that for a non-primitive proof $\mathfrak{R}(\iota)$ of some S proof ι, where S is one of our four QCDCL variants, ι needs to consist of a trail that assigns a U-literal out-of-order (i.e., before we have assigned all X-literals).

Since neither cube learning nor pure literal elimination is allowed in QCDCL, we can immediately conclude:

Corollary 1. *Let ι be a* QCDCL *refutation of a QCNF Φ that fulfils the XT-property. Then $\mathfrak{R}(\iota)$ is primitive.*

We remark that some of the QBFs we introduce in the paper are not minimally false, i.e., we have added extra clauses to formulas that were false already. Although this is unusual in proof complexity, practical (false) instances are not guaranteed to be minimally false. Therefore it is natural to also consider these QBFs when investigating QCDCL systems. These algorithmic proof systems have to utilize all clauses, even if they are redundant for Q-resolution refutations.

5 Plain QCDCL vs. QCDCL with Cubes/PL

We start by examining the influence of cube learning on our QCDCL model. For false formulas we can always prevent learning cubes by just deciding the universal variables according to a winning strategy for the universal player, which will cause a conflict on the current trail. Thus cube learning will never be disadvantageous in principle.

Proposition 2. QCDCL^{CUBE} *p-simulates* QCDCL *and* QCDCL$^{CUBE+PL}$ *p-simulates* QCDCL^{PL}*.*

We recall the equality formulas $\texttt{Eq}_n$ of Beyersdorff *et al.* [2019a]. These are QCNFs with prefix $\exists x_1 \dots x_n \forall u_1 \dots u_n \exists t_1 \dots t_n$ and matrix

$$(\bar{t}_1 \vee \dots \vee \bar{t}_n) \wedge \bigwedge_{i=1}^{n} ((\bar{x}_i \vee \bar{u}_i \vee t_i) \wedge (x_i \vee u_i \vee t_i)).$$

The formulas are known to be hard for Q-resolution [Beyersdorff *et al.*, 2019a] and also for QCDCL [Beyersdorff and Böhm, 2021]. In contrast, we show that they are easy in QCDCL with cube learning.

Proposition 3. *There exist polynomial-size* QCDCL^{CUBE} *refutations of* $\texttt{Eq}_n$*.*

Proof Sketch. First we learn the cubes $x_i \wedge \bar{u}_i$ and $\bar{x}_i \wedge u_i$ for $i \in [n-1]$. E.g., to learn $x_1 \wedge \bar{u}_1$, we use the trail

$$\mathcal{T}_1 := (\mathbf{x_1}; \dots; \mathbf{x_n}; \bar{\mathbf{u}}_\mathbf{1}; \dots; \bar{\mathbf{u}}_\mathbf{n}; \bar{\mathbf{t}}_\mathbf{1}; \mathbf{t_2}; \dots; \mathbf{t_n}).$$

Then the partial assignment $x_1 \wedge \bar{u}_1 \wedge \bar{t}_1 \wedge t_2 \wedge \dots \wedge t_n$ satisfies the matrix of $\texttt{Eq}_n$. Reducing this cube existentially results in $x_1 \wedge \bar{u}_1$, hence $x_1 \wedge \bar{u}_1 \in \mathfrak{L}(\mathcal{T}_1)$.

Having learnt all these $2n-2$ cubes, we learn the clauses $L_i := \bar{x}_i \vee \bar{u}_i \vee \bigvee_{j=i+1}^{n}(u_j \vee \bar{u}_j) \vee \bigvee_{k=1}^{i-1} \bar{t}_k$ and $R_i := x_i \vee u_i \vee \bigvee_{j=i+1}^{n}(u_j \vee \bar{u}_j) \vee \bigvee_{k=1}^{i-1} \bar{t}_k$, starting with $i = n-1$ and down to $i = 2$. E.g., to learn L_{n-1} we use the trail

$$\mathcal{U}_{n-1} := (\mathbf{x_1}, u_1, t_1; \dots; \mathbf{x_{n-1}}, u_{n-1}, t_{n-1}, \bar{t}_n, x_n, \bot),$$

where u_i are propagated directly after x_i by the learnt cubes. We resolve over $x_n, \bar{t}_n$, and t_{n-1} to get L_{n-1}.

Having finally learnt L_2 and R_2, we form the trail $\mathcal{U}_1 := (\mathbf{x_1}, u_1, t_1, x_2, \bot)$ with $\text{ante}_{\mathcal{U}_1}(x_2) = R_2$ and $\text{ante}_{\mathcal{U}_1}(\bot) = L_2$ and learn $(\bar{x}_1)$. Then the last trail $(\bar{x}_1, \bar{u}_1, t_1, x_2, \bot)$ yields the empty clause. $\square$

As the formulas $\texttt{Eq}_n$ require exponential-sized QCDCL refutations [Beyersdorff and Böhm, 2021] we obtain:

Theorem 3. QCDCL^{CUBE} *is exponentially stronger than* QCDCL*.*

Next we will look at the influence of pure literal elimination. Now, the effect of pure literal elimination is similar to cube learning: they enable out-of-order decisions that can shorten the refutations. This again manifests in $\texttt{Eq}_n$.

Proposition 4. $\texttt{Eq}_n$ *has poly-size* QCDCL^{PL} *(and* QCDCL$^{CUBE+PL}$*) refutations.*

Although pure literal elimination helps to refute $\texttt{Eq}_n$, it turns out that pure literal elimination can also be disadvantageous. It might be a fallacy to think that pure existential literals should be satisfied in the same way as unit clauses in unit propagation. We will construct formulas in which pure literal elimination thwarts finding a convenient conflict and therefore short refutations.

We construct these formulas in stages, starting with $\texttt{MirrorCR}_n$. In turn, these QBFs are based on the Completion Principle $\texttt{CR}_n$ of Janota [2016], known to be hard for QCDCL [Janota, 2016; Böhm and Beyersdorff, 2021]. The "Mirror"-modification adds new symmetries to the formula, causing pure literals to appear too late to make a difference.

Definition 7. *The QCNF* $\texttt{MirrorCR}_n$ *consists of the prefix* $\exists x_{(1,1)}, \dots, x_{(n,n)} \forall u \exists a_1, \dots, a_n, b_1, \dots, b_n$ *and the matrix* $x_{(i,j)} \vee u \vee a_i$, $\bar{a}_1 \vee \dots \vee \bar{a}_n$, $\bar{x}_{(i,j)} \vee \bar{u} \vee b_j$, $\bar{b}_1 \vee \dots \vee \bar{b}_n$ $x_{(i,j)} \vee \bar{u} \vee \bar{a}_i$, $a_1 \vee \dots \vee a_n$, $\bar{x}_{(i,j)} \vee u \vee \bar{b}_j$, $b_1 \vee \dots \vee b_n$ *for $i, j \in [n]$.*

It is easy to see that $\texttt{MirrorCR}_n$ fulfil the XT-property. Additionally, we can show:

Proposition 5. *The CNF* $\mathfrak{C}(\texttt{MirrorCR}_n)$ *is unsatisfiable and gauge*$(\texttt{MirrorCR}_n) \geqslant n-1$.

Applying Theorem 2 we infer:

Corollary 2. $\texttt{MirrorCR}_n$ *needs exponential-size fully reduced primitive* Q-resolution *refutations.*

All we have to do now is to show that all $\mathsf{QCDCL}^{\mathsf{PL}}$ refutations of $\texttt{MirrorCR}_n$ are primitive. Then the gauge lower bound applies. We will show that for a non-primitive refutation of $\texttt{MirrorCR}_n$ we would need to decide literals by pure literal elimination, and before each pure literal elimination we have to perform another one, which is a contradiction.

Proposition 6. *From each* $\mathsf{QCDCL}^{\mathsf{PL}}$ *refutation of* $\texttt{MirrorCR}_n$ *we can extract a fully reduced primitive* Q-resolution *refutation of the same size.*

Proof Sketch. We show that $\mathfrak{R}(\iota)$ is primitive if ι is a $\mathsf{QCDCL}^{\mathsf{PL}}$ refutation of $\texttt{MirrorCR}_n$. Assume that some $\mathfrak{R}(\iota)$ is not primitive. Therefore, by Proposition 1, a U-literal was decided as a pure literal before all X-variables were assigned. However, such a situation is impossible due to the formula structure, resulting in a contradiction. □

Corollary 3. *The QBFs* $\texttt{MirrorCR}_n$ *require exponential-size* $\mathsf{QCDCL}^{\mathsf{PL}}$ *refutations.*

Next we embed this formula into a new QCNF $\texttt{PLTrap}_n$.

Definition 8. *The QCNF* $\texttt{PLTrap}_n$ *has the prefix* $\exists y, x_{(1,1)}, \dots, x_{(n,n)} \forall u \exists a_1, \dots, a_n, b_1, \dots, b_n, a, b$. *Its matrix contains all clauses from* $\texttt{MirrorCR}_\texttt{n}$ *and additionally* $(y \vee a)$, $(\bar{a} \vee b)$, $(\bar{a} \vee \bar{b})$, $(a \vee b)$, *and* $(a \vee \bar{b})$.

Proposition 7. $\texttt{PLTrap}_n$ *needs exponential-size* $\mathsf{QCDCL}^{\mathsf{PL}}$ *and* $\mathsf{QCDCL}^{\mathsf{CUBE+PL}}$ *refutations.*

Not having to follow the PLD rule, we can construct short proofs of $\texttt{PLTrap}_n$ by focusing on the new clauses over a, b.

Proposition 8. $\texttt{PLTrap}_n$ *has polynomial-size* QCDCL *refutations.*

We conclude that pure literal elimination is advantageous for $\texttt{Eq}_n$, but not for $\texttt{PLTrap}_n$. Therefore we obtain:

Theorem 4. $\mathsf{QCDCL}^{\mathsf{PL}}$ *and* QCDCL *are incomparable as well as* $\mathsf{QCDCL}^{\mathsf{CUBE+PL}}$ *and* QCDCL.

6 Cube Learning vs. Pure Literal Elimination

As shown in Section 5, cube learning improves QCDCL, while adding pure literal elimination leads to incomparable systems. Thus it is natural to directly compare cube learning and pure literal elimination. Because of the results above, we cannot use $\texttt{Eq}_n$ for a potential separation. However, we can modify the QBFs such that they remain easy for $\mathsf{QCDCL}^{\mathsf{PL}}$, while eliminating the benefits from cube learning.

Definition 9. *The QCNF* $\texttt{TwinEq}_n$ *has the prefix* $\exists x_1, \dots, x_n \forall u_1, \dots, u_n, w_1, \dots, w_n \exists t_1, \dots, t_n$. *Its matrix contains the clauses from* $\texttt{Eq}_n$ *together with* $x_i \vee w_i \vee t_i$ *and* $\bar{x}_i \vee \bar{w}_i \vee t_i$ *for* $i \in [n]$.

The main idea of this twin construction is to ensure that all potential cubes consist of at least two universal variables. We can do the same construction for other QCNFs.

Obviously, $\texttt{TwinEq}_n$ fulfils the XT-property. We compute gauge$(\texttt{TwinEq}_n) = n$ and hence infer by Theorem 2:

Proposition 9. *Fully reduced primitive* Q-resolution *refutations of* $\texttt{TwinEq}_n$ *have exponential size.*

We show that $\mathsf{QCDCL}^{\mathsf{CUBE}}$ refutations of $\texttt{TwinEq}_n$ are primitive by proving that it is impossible to propagate U-literals before having assigned all X-literals.

Proposition 10. *Each* $\mathsf{QCDCL}^{\mathsf{CUBE}}$ *refutation of* $\texttt{TwinEq}_n$ *has at least exponential size.*

Having shown that $\texttt{TwinEq}_n$ is hard for $\mathsf{QCDCL}^{\mathsf{CUBE}}$, it remains to prove that it is easy for $\mathsf{QCDCL}^{\mathsf{PL}}$.

Proposition 11. $\texttt{TwinEq}_n$ *has polynomial-size* $\mathsf{QCDCL}^{\mathsf{PL}}$ *refutations.*

For the other separation we use $\texttt{PLTrap}_n$, which is hard for $\mathsf{QCDCL}^{\mathsf{PL}}$, but still easy for $\mathsf{QCDCL}^{\mathsf{CUBE}}$ by Proposition 2. Therefore we conclude:

Theorem 5. $\mathsf{QCDCL}^{\mathsf{CUBE}}$ *is incomparable to* $\mathsf{QCDCL}^{\mathsf{PL}}$.

We have seen earlier that the QCDCL system including pure literal elimination is incomparable to the system without. Now we will prove that this incomparability still holds with cube learning turned on. By Proposition 2, we obtain that $\mathsf{QCDCL}^{\mathsf{CUBE+PL}}$ p-simulates $\mathsf{QCDCL}^{\mathsf{PL}}$. Therefore we get from Proposition 11:

Corollary 4. $\texttt{TwinEq}_n$ *has poly-size* $\mathsf{QCDCL}^{\mathsf{CUBE+PL}}$ *refutations.*

Since $\texttt{TwinEq}_n$ is hard for $\mathsf{QCDCL}^{\mathsf{CUBE}}$, this gives us the first separation between $\mathsf{QCDCL}^{\mathsf{CUBE+PL}}$ and $\mathsf{QCDCL}^{\mathsf{CUBE}}$. The other direction can be shown with $\texttt{PLTrap}_n$.

Proposition 12. $\texttt{PLTrap}_n$ *has poly-size* $\mathsf{QCDCL}^{\mathsf{CUBE}}$ *refutations.*

Hence we get:

Theorem 6. $\mathsf{QCDCL}^{\mathsf{CUBE+PL}}$ *and* $\mathsf{QCDCL}^{\mathsf{CUBE}}$ *are incomparable.*

We now consider the relation between $\mathsf{QCDCL}^{\mathsf{CUBE+PL}}$ and $\mathsf{QCDCL}^{\mathsf{PL}}$. We introduce another modification of $\texttt{Eq}_n$, which we call $\texttt{BulkyEq}_n$, where we add two clauses.

Definition 10. *The QCNF* $\texttt{BulkyEq}_n$ *is obtained from* $\texttt{Eq}_n$ *by adding the clauses* $u_1 \vee \dots \vee u_n \vee t_1 \vee \dots \vee t_n$ *and* $\bar{u}_1 \vee \dots \vee \bar{u}_n \vee t_1 \vee \dots \vee t_n$ *to the matrix.*

As $\texttt{Eq}_n$, this formula fulfils the XT-property and has a high gauge value ($\geqslant n-1$). By Theorem 2 we infer that $\texttt{BulkyEq}_n$ needs exponential-size fully reduced primitive Q-resolution refutations. Similarly to $\texttt{MirrorCR}_n$, we can then show that pure literal elimination does not shorten proofs because of the two additional 'bulky' clauses that prevent pure literals to occur early in trails. Therefore $\texttt{BulkyEq}_n$ is hard for $\mathsf{QCDCL}^{\mathsf{PL}}$. On the other hand, we can explicitly construct short proofs in $\mathsf{QCDCL}^{\mathsf{CUBE+PL}}$. Therefore we get:

Proposition 13. $\texttt{BulkyEq}_n$ *has poly-size* $\mathsf{QCDCL}^{\mathsf{CUBE+PL}}$ *refutations, but needs exponential-size* $\mathsf{QCDCL}^{\mathsf{PL}}$ *refutations.*

As for the systems without pure literal elimination, we get:

Theorem 7. $\mathsf{QCDCL}^{\mathsf{CUBE+PL}}$ *is exponentially stronger than* $\mathsf{QCDCL}^{\mathsf{PL}}$.

7 The QCDCL Systems vs. Q-resolution

Beyersdorff and Böhm [2021] showed incomparability of Q-resolution and QCDCL. We now lift this to the other QCDCL variants introduced here. For one separation, we can use the QCNF $\texttt{QParity}_n$ from [Beyersdorff *et al.*, 2019b], which have short QCDCL refutations. These formulas have prefix $\exists x_1 \dots x_n \forall u \exists t_1 \dots t_n$ and clauses

$$\begin{gathered} x_1 \vee \bar{t}_1,\ \ \bar{x}_1 \vee t_1,\ \ u \vee t_n,\ \ \bar{u} \vee \bar{t}_n, \\ x_i \vee t_{i-1} \vee \bar{t}_i,\ \ x_i \vee \bar{t}_{i-1} \vee t_i, \\ \bar{x}_i \vee t_{i-1} \vee t_i,\ \ \bar{x}_i \vee \bar{t}_{i-1} \vee \bar{t}_i \quad \text{for } i \in \{2, \dots, n\}. \end{gathered}$$

Theorem 8. QCDCL, $\mathsf{QCDCL}^{\mathsf{CUBE}}$, $\mathsf{QCDCL}^{\mathsf{PL}}$ *and* $\mathsf{QCDCL}^{\mathsf{CUBE+PL}}$ *are all incomparable to* Q-resolution.

In detail, the QBFs $\texttt{QParity}_n$ *have polynomial-size* QCDCL, $\mathsf{QCDCL}^{\mathsf{CUBE}}$, $\mathsf{QCDCL}^{\mathsf{PL}}$, *and* $\mathsf{QCDCL}^{\mathsf{CUBE+PL}}$ *refutations, but need exponential-size* Q-resolution *refutations. On the other hand,* $\texttt{MirrorCR}_n$ *have polynomial-size* Q-resolution *refutations, but need exponential-size* QCDCL, $\mathsf{QCDCL}^{\mathsf{CUBE}}$, $\mathsf{QCDCL}^{\mathsf{PL}}$, *and* $\mathsf{QCDCL}^{\mathsf{CUBE+PL}}$ *refutations.*

8 Experiments

We study proof systems in the hope that proof-complexity results will translate to running-time complexity for solvers. In this section we do our reality check to see whether this hypothesis holds up experimentally.

We picked the solver DepQBF [Lonsing and Egly, 2017], as it is the only one that supports pure-literal elimination (PLE) and also has the ability to turn cube learning (SDCL) off.[1] We additionally ran Qute [Peitl *et al.*, 2019] when we wanted to confirm DepQBF's surprising behaviour. Though, Qute only supports $\mathsf{QCDCL}^{\mathsf{CUBE}}$ of the systems discussed above, and so is poorly suited for most of our experiments.

We ran DepQBF on each of the formulas used for separations in this paper, as well as on the PCNF track of the QBF Evaluation 2020.[2] We set the time limit on each formula to 10 minutes (no memory limit). The computation was performed on a machine with two 16-core Intel® Xeon® E5-2683 v4@2.10GHz CPUs and 512GB RAM running Ubuntu 20.04.3 LTS on Linux 5.4.0-48, and organized with the help of GNU Parallel [Tange, 2011].

We discovered that *heuristics* have a significant impact on performance on theoretically easy formulas (theoretically hard formulas must be hard for solvers as well, and we confirm this in every case). We thus decided to run DepQBF

[1]In order to obtain vanilla QCDCL in DepQBF, we set `--traditional-qcdcl --long-dist-res --dep-man=simple --no-dynamic-nenofex --no-trivial-truth --no-trivial-falsity`.

[2]http://www.qbflib.org/qbfeval20.php

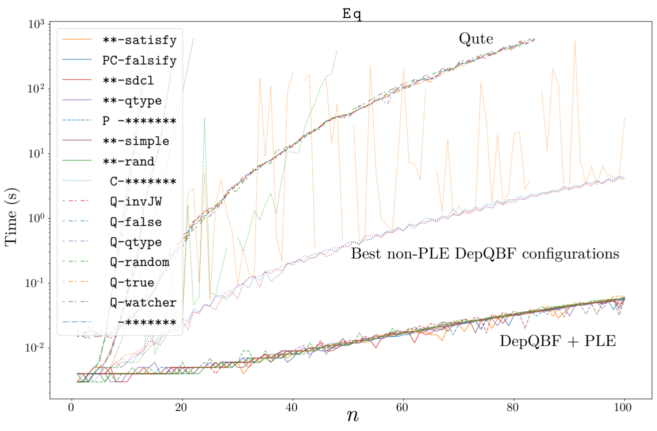

Figure 2: Labels indicate whether PLE (`P*-`) and SDCL (`*C-`) are on, configurations of one kind have the same line style. Lines for Qute start with `Q-`, the remaining lines are for DepQBF. The rest of the label is the heuristic; configurations with the same heuristic share colour. Gaps in lines indicate time-outs at 10 minutes. The legend is sorted in descending order of performance; some labels are omitted to save space (at least one of each colour and line style is shown).

with each available heuristic, in order to paint a full picture. In total, we evaluated 24 configurations of DepQBF—with and without PLE and with and without SDCL, and for each of these four, with each of the 6 possible heuristics.[3]

Figure 2 shows the results on Equality. While the formulas are easy using PLE regardless of the heuristic, without PLE DepQBF's performance fluctuates depending on the heuristic, even though the formulas are 'easy' as long as SDCL is on. Qute's performance is stabler, but still much worse than DepQBF with PLE. The formulas get hard without both PLE and SDCL, in line with [Beyersdorff and Böhm, 2021].

While both PLE and SDCL make `Eq` easy, PLE seems easier to exploit. Perhaps there is simply less room for error in applying PLE than in learning the right cubes. A mild advantage from PLE is also confirmed by DepQBF's results on QBF Eval formulas. This suggests that in spite of conceptual simplicity PLE can be quite useful, and perhaps should appear on Qute's feature roadmap.

9 Conclusion

While this paper only studies false formulas (in accordance with proof complexity conventions), we expect similar phenomena of incomparability on true formulas, which we leave for future work to explore. Interestingly, while cube learning is primarily needed for true QBFs, we have shown that it can also improve the running time on false instances.

Technically, we believe that our new notion of primitive proofs has further potential for showing QCDCL lower bounds, also for QBFs of higher quantifier complexity. While previous results tried to lift lower bounds from Q-Resolution [Beyersdorff and Böhm, 2021], primitivity also applies to QBFs easy for Q-Resolution, thus supplying new reasons for QCDCL hardness.

[3]Using `--dec-heur=` (`--phase-heuristic` for Qute).

Acknowledgements

Research was supported by grants from the Carl-Zeiss Foundation, Austrian Science Fund FWF grant no. J4361-N, and DFG grant BE 4209/3-1.

References

[Atserias *et al.*, 2011] Albert Atserias, Johannes Klaus Fichte, and Marc Thurley. Clause-learning algorithms with many restarts and bounded-width resolution. *J. Artif. Intell. Res.*, 40:353–373, 2011.

[Balabanov and Jiang, 2012] Valeriy Balabanov and Jie-Hong R. Jiang. Unified QBF certification and its applications. *Form. Methods Syst. Des.*, 41(1):45–65, 2012.

[Beame *et al.*, 2004] Paul Beame, Henry A. Kautz, and Ashish Sabharwal. Towards understanding and harnessing the potential of clause learning. *J. Artif. Intell. Res. (JAIR)*, 22:319–351, 2004.

[Beyersdorff and Böhm, 2021] Olaf Beyersdorff and Benjamin Böhm. Understanding the Relative Strength of QBF CDCL Solvers and QBF Resolution. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:20, 2021.

[Beyersdorff *et al.*, 2019a] Olaf Beyersdorff, Joshua Blinkhorn, and Luke Hinde. Size, cost, and capacity: A semantic technique for hard random QBFs. *Logical Methods in Computer Science*, 15(1), 2019.

[Beyersdorff *et al.*, 2019b] Olaf Beyersdorff, Leroy Chew, and Mikolás Janota. New resolution-based QBF calculi and their proof complexity. *ACM Transactions on Computation Theory*, 11(4):26:1–26:42, 2019.

[Beyersdorff *et al.*, 2021] Olaf Beyersdorff, Mikolás Janota, Florian Lonsing, and Martina Seidl. Quantified boolean formulas. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, Frontiers in Artificial Intelligence and Applications, pages 1177–1221. IOS Press, 2021.

[Biere *et al.*, 2021a] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, Frontiers in Artificial Intelligence and Applications. IOS Press, 2021.

[Biere *et al.*, 2021b] Armin Biere, Matti Järvisalo, and Benjamin Kiesl. Preprocessing in sat solving. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, Frontiers in Artificial Intelligence and Applications, pages 291–436. IOS Press, 2021.

[Böhm and Beyersdorff, 2021] Benjamin Böhm and Olaf Beyersdorff. Lower bounds for QCDCL via formula gauge. In *Theory and Applications of Satisfiability Testing (SAT 2021)*, pages 47–63. Springer, 2021.

[Buss and Nordström, 2021] Sam Buss and Jakob Nordström. Proof complexity and sat solving. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, Frontiers in Artificial Intelligence and Applications, pages 233–350. IOS Press, 2021.

[Elffers *et al.*, 2018] Jan Elffers, Jesús Giráldez-Cru, Stephan Gocht, Jakob Nordström, and Laurent Simon. Seeking practical CDCL insights from theoretical SAT benchmarks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1300–1308. ijcai.org, 2018.

[Janota, 2016] Mikolás Janota. On Q-Resolution and CDCL QBF solving. In *Proc. International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 402–418, 2016.

[Kleine Büning *et al.*, 1995] Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. Resolution for quantified Boolean formulas. *Inf. Comput.*, 117(1):12–18, 1995.

[Kokkala and Nordström, 2020] Janne I. Kokkala and Jakob Nordström. Using resolution proofs to analyse CDCL solvers. In *Principles and Practice of Constraint Programming - 26th International Conference (CP)*, pages 427–444. Springer, 2020.

[Krajíček, 2019] Jan Krajíček. *Proof complexity*, volume 170 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, 2019.

[Lonsing and Egly, 2017] Florian Lonsing and Uwe Egly. DepQBF 6.0: A search-based QBF solver beyond traditional QCDCL. In *Proc. International Conference on Automated Deduction (CADE)*, pages 371–384, 2017.

[Lonsing, 2012] Florian Lonsing. *Dependency Schemes and Search-Based QBF Solving: Theory and Practice*. PhD thesis, Johannes Kepler University Linz, 2012.

[Marques Silva *et al.*, 2021] João P. Marques Silva, Inês Lynce, and Sharad Malik. Conflict-driven clause learning SAT solvers. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, Frontiers in Artificial Intelligence and Applications. IOS Press, 2021.

[Peitl *et al.*, 2019] Tomás Peitl, Friedrich Slivovsky, and Stefan Szeider. Dependency learning for QBF. *J. Artif. Intell. Res.*, 65:180–208, 2019.

[Pipatsrisawat and Darwiche, 2011] Knot Pipatsrisawat and Adnan Darwiche. On the power of clause-learning SAT solvers as resolution engines. *Artif. Intell.*, 175(2):512–525, 2011.

[Sakallah and Marques-Silva, 2011] Karem A. Sakallah and João Marques-Silva. Anatomy and empirical evaluation of modern SAT solvers. *Bull. EATCS*, 103:96–121, 2011.

[Tange, 2011] Ole Tange. GNU parallel: The command-line power tool. *login Usenix Mag.*, 36(1), 2011.

[Vinyals, 2020] Marc Vinyals. Hard examples for common variable decision heuristics. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[Zhang and Malik, 2002] Lintao Zhang and Sharad Malik. Conflict driven learning in a quantified Boolean satisfiability solver. In *Proc. IEEE/ACM International Conference on Computer-aided Design (ICCAD)*, pages 442–449, 2002.