Parameterized Bounded-Depth Frege is Not Optimal

Olaf Beyersdorff[*]
Institut für Theoretische Informatik
Leibniz Universität Hannover, Germany

Nicola Galesi[†] Massimo Lauria
Dipartimento di Informatica
Sapienza Università di Roma, Italy

Alexander Razborov[‡]
Department of Computer Science
The University of Chicago

Abstract

A general framework for parameterized proof complexity was introduced by Dantchev, Martin, and Szeider [7]. In that framework the parameterized version of any proof system is not fpt-bounded for some technical reasons, but we remark that this question becomes much more interesting if we restrict ourselves to those parameterized contradictions (F, k) in which F itself is a contradiction. We call such parameterized contradictions *strong*, and with one important exception (vertex cover) all interesting contradictions we are aware of are strong. It follows from the gap complexity theorem of [7] that tree-like Parameterized Resolution is not fpt-bounded w.r.t. strong parameterized contradictions.

The main result of this paper significantly improves upon this by showing that even the parameterized version of bounded-depth Frege is not fpt-bounded w.r.t. strong contradictions. More precisely, we prove that the pigeonhole principle requires proofs of size $n^{\Omega(k)}$ in bounded-depth Frege, and, as a special case, in dag-like Parameterized Resolution. This answers an open question posed in [7].

In the opposite direction, we interpret a well-known FPT algorithm for vertex cover as a DPLL procedure for Parameterized Resolution. Its generalization leads to a proof search algorithm for Parameterized Resolution that in particular shows that tree-like Parameterized Resolution allows short refutations of all parameterized contradictions given as bounded-width CNF's.

[*]Part of this work was done while the first author was visiting Sapienza University Rome under support of DFG grant KO 1053/5–2.

[†]Supported by grant "Limiti di compressione in combinatoria e complessità computazionale" by Sapienza University Rome.

[‡]Part of this work was done while the author was at Steklov Mathematical Institute, supported by the Russian Foundation for Basic Research, and at Toyota Technological Institute, Chicago.

1 Introduction

Recently, Dantchev, Martin, and Szeider [7] introduced the framework of *parameterized proof complexity*, an extension of the proof complexity approach of Cook and Reckhow to parameterized complexity. One motivation for this is the quest for efficient algorithms which solve optimization problems [8, 9, 14]. Since Resolution is very important for SAT solving, its analogue in this context, Parameterized Resolution, combines these two approaches, and its investigation might provide new insights into proof search for tractable fragments of classically hard problems. Some results in this direction are already outlined in the work of Gao [11] where he analyzes the effect of the standard DPLL algorithm on the problem of weighted satisfiability for random d-CNF. However, the study of Parameterized Resolution and our understanding of the possible implications for SAT-solving algorithms are still at a very early stage.

More generally, *parameterized complexity* is a branch of complexity theory where problems are analyzed in a finer way than in the classical approach: we say that a problem is *fixed-parameter tractable* (FPT) with parameter k if it can be solved in time $f(k)n^{O(1)}$ for some computable function f of arbitrary growth. In this setting, classically intractable problems may have efficient solutions for small choices of the parameter, even if the total size of the input is large. Parameterized complexity also has completeness theory under a suitably modified notion of a polynomial reduction called an *fpt-reduction*. Many parameterized problems that appear to be not fixed-parameter tractable have been classified as being complete under fpt-reductions for complexity classes in the so-called weft hierarchy $\mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \mathsf{W}[3] \subseteq \dots$

Consider the problem WEIGHTED CNF SAT of finding a truth assignment of Hamming weight at most k that satisfies all clauses of a formula in conjunctive normal form. Many parameterized combinatorial problems can be naturally encoded in WEIGHTED CNF SAT: finding a vertex cover of size at most k, finding a clique of size k, or finding a dominating set of size at most k. In the theory of parameterized complexity, the hardness of the WEIGHTED CNF SAT problem is reflected by its $\mathsf{W}[2]$-completeness. Parameterized complexity is a very well-developed and deep theory and, as for the classical case, there are many open problems concerning the separation of parameterized complexity classes as FPT and $\mathsf{W}[\mathsf{P}]$ (see [8, 10, 14] for a comprehensive treatment of the field).

After considering the notions of propositional *parameterized tautologies* and *fpt-bounded* proof systems, Dantchev, Martin, and Szeider [7] laid the foundations to study complexity of proofs in a parameterized setting. The problem WEIGHTED CNF SAT leads to parameterized contradictions:

Definition 1 (see [7]). *A* parameterized contradiction *is a pair (F, k) consisting of a propositional formula F and $k \in \mathbb{N}$ such that F has no satisfying assignment of weight $\leq k$. We denote the set of all parameterized contradictions by* PCon*.*

The notions of a parameterized proof system and of fpt-bounded proof systems were also developed in [7] (see Section 2 for a discussion):

Definition 2 (Dantchev et al. [7]). *A* parameterized proof system *for a parameterized language $L \subseteq \Sigma^* \times \mathbb{N}$ is a function $P : \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$ such that $rng(P) = L$ and $P(x, k)$ can be computed in time $O(f(k)|x|^{O(1)})$ with some computable function f.*

The system P is fpt-bounded *if there exist computable functions s and t such that every $(x, k) \in L$ has a P-proof (y, k') with $|y| \leq s(k)|x|^{O(1)}$ and $k' \leq t(k)$.*

The main motivation behind the work of [7] was that of generalizing the classical approach of Cook and Reckhow to the parameterized case and working towards a separation of parameterized complexity classes as FPT and W[P] by techniques developed in proof complexity. In fact, we obtain an analogous result to the well-known Cook-Reckhow theorem from [6]:

Theorem 3. *A parameterized language $L \subseteq \Sigma^* \times \mathbb{N}$ has an fpt-bounded proof system if and only if $L \in \mathsf{para\text{-}NP}$.*

In [7] (tree-like) *Parameterized Resolution* was defined as a refutation system for the set of parameterized contradictions. Given a set of clauses F in variables $x_1, \dots, x_n$ with $(F, k) \in \text{PCon}$, a (tree-like) *Parameterized Resolution refutation* of (F, k) is a (tree-like) Resolution refutation of $F \cup \{\neg x_{i_1} \vee \dots \vee \neg x_{i_{k+1}} \mid 1 \leq i_1 < \dots < i_{k+1} \leq n\}$. Thus, in (tree-like) Parameterized Resolution we have built-in access to all parameterized clauses of the form $\neg x_{i_1} \vee \dots \vee \neg x_{i_{k+1}}$. All these clauses are available in the system, but when measuring the size of a derivation we only count those which actually appear in the derivation.

This concept can be straightforwardly generalized to an arbitrary proof system P, be it dag-like or tree-like, that understands clauses and works with lines. However, as was indicated by a referee of the previous version of this paper, parameterized proof systems are never fpt-bounded for some technical reasons (see Example 1 in Section 2.3). We remark that disappointing examples of this sort seem to necessarily stem from the fact that the formula F in Definition 4 is *not* a contradiction by itself. Let us call a parameterized contradiction (F, k) *strong* if F itself is a contradiction (we will lay down our arguments as to why, in our opinion, this notion is interesting in the same Section 2.3). Let us call a parameterized proof system P for the language PCon *weakly fpt-bounded* if the existence of a good P-proof (y, k') in Definition 2 is guaranteed only when the parameterized contradiction (x, k) is strong.

Dantchev et al. [7] prove an extension of Riis' *gap theorem* [18], getting a classification for the complexity of tree-like Parameterized Resolution proofs for parameterized contradictions arising from propositional encodings of first-order principles $\mathcal{P}$, that uniquely depends on $\mathcal{P}$ having or not infinite models. As all parameterized contradictions obtainable in this way are strong, their main result implied that tree-like Parameterized Resolution is not weakly fpt-bounded. A similar question for dag-like Parameterized Resolution was left open in [7]. More specifically, they asked if (the parameterized version of) the pigeonhole principle is hard for dag-like Parameterized Resolution.

1.1 Our Contributions

We answer this question by proving that PHP^{n+1}_n requires proofs of size $n^{\Omega(k)}$ not only in Parameterized Resolution but in the much stronger system of bounded-depth Frege. This in particular implies that bounded-depth Frege is not even weakly fpt-bounded. Our result is in sharp contrast with [7, Proposition 17] that gave efficient proofs of PHP^{n+1}_n in Parameterized Resolution using a more sophisticated encoding with auxiliary variables, and we discuss these augmented proof systems in the final Section 5.

As our second contribution we investigate classes of parameterized contradictions that have short refutations in tree-like Parameterized Resolution. The notion of *efficient kernelization* plays an important role in the theory of parameterized complexity to design fpt-algorithms. Here we propose a notion of kernel for parameterized proof complexity: a parameterized contradiction (F, k) has a kernel if there exists a subset of clauses $F' \subseteq F$ whose size is bounded by a function of k only and such that (F', k) is still a parameterized contradiction. We observe that if a formula has a kernel, then it can be efficiently refuted in tree-like Parameterized Resolution. As an immediate consequence several examples of formulas hard for tree-like Resolution are instead efficiently refutable in the parameterized case: pebbling contradictions, linear ordering principles, graph pigeonhole principles, and colorability principles. But sometimes a kernel of a formula is not explicit or immediate to find. In Theorem 15 we prove that contradictions of bounded width have a kernel and thus very efficient tree-like Parameterized Resolution refutations.

Is the existence of a kernel a necessary condition for a parameterized contradiction to have an fpt-bounded refutation in tree-like Parameterized resolution? A trivial counterexample to this conjecture is made by the CNF $(x_1 \vee x_2 \vee \ldots \vee x_n) \wedge \neg x_1 \wedge \ldots \wedge \neg x_n$. We also include a much more interesting example (Theorem 18) of a parameterized contradiction, a version of the linear ordering principle, that has fpt-refutations in tree-like Parameterized Resolution without having a kernel.

1.2 Techniques and Proof Methods

Our lower bound for the pigeonhole principle is a rather simple application of the method of random restrictions introduced in proof complexity by Haken in his seminal paper [12]. But our choice of parameters is totally different and allows us to kill with the restriction any small prescribed set of parameterized axioms. While the technique is routine, it nonetheless seems to be its first application in the context of *parameterized* complexity, be it computational or proof complexity.

Gao [11] suggested to use a standard DPLL algorithm to find refutations of certain random parameterized d-CNF's. Here we prove that bounded width CNF's have a kernel and hence are efficiently refutable in tree-like Parameterized Resolution (Theorem 15). The core of our argument is the interpretation of a classical parameterized algorithm for vertex cover as a DPLL procedure. This results in a very simple algorithm.

1.3 Organization of the Paper

The remaining part of the paper is organized as follows. Section 2 contains all preliminary notions and definitions concerning fixed-parameter tractability, parameterized proof systems, and Parameterized Resolution. In Section 3 we show that general Parameterized Resolution is not fpt-bounded by proving a lower bound for the pigeonhole principle. Section 4 concentrates on upper bounds: we introduce the notion of a kernel and prove that parameterized contradictions of bounded width have efficient tree-like refutations. We also present a variant of the linear ordering principle that possesses an efficient tree-like refutation but does not have a kernel. We conclude in Section 5 with a brief discussion, an outline of future directions and some open problems.

2 Parameterized Proof Complexity

2.1 Fixed-Parameter Tractability

A *parameterized language* is a language $L \subseteq \Sigma^* \times \mathbb{N}$. For an instance (x, k), we call k the *parameter of* (x, k). A parameterized language L is *fixed-parameter tractable* if L has a deterministic decision algorithm running in time $f(k)|x|^{O(1)}$ for some computable function f. The class of all fixed-parameter tractable languages is denoted by FPT.

Besides FPT there is a wealth of complexity classes containing problems which are not believed to be fixed-parameter tractable. The most prominent classes lie in the *weft hierarchy* forming a chain

$$\mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \cdots \subseteq \mathsf{W}[\mathsf{P}] \subseteq \mathsf{para\text{-}NP} \ .$$

The classes of the weft hierarchy are usually defined as the closure of a canonical problem under fpt-reductions. For $\mathsf{W}[2]$ this canonical problem is Weighted CNF Sat containing instances (F, k) with a propositional formula F in CNF and a parameter $k \in \mathbb{N}$. Weighted CNF Sat asks whether F has a satisfying assignment of weight k, where the weight of an assignment α, denoted $w(\alpha)$, is the number of variables that α assigns to 1. Instead of asking for an assignment α with $w(\alpha) = k$ we can also ask for α with $w(\alpha) \leq k$ and still get a $\mathsf{W}[2]$-complete problem (cf. [7]). Like in the classical duality between tautologies and satisfiability, we obtain a complete problem for $\mathsf{coW}[2]$:

Definition 4 (Dantchev, Martin, Szeider [7])**.** *A* parameterized contradiction *is a pair (F, k) consisting of a propositional formula F, given as a CNF, and $k \in \mathbb{N}$ such that F has no satisfying assignment of weight $\leq k$. We denote the set of all parameterized contradictions by* PCon.

For an in-depth treatment of notions from parameterized complexity we refer to the monograph [8, 10, 14].

2.2 Parameterized Proof Systems

We start with discussing the general definition of a parameterized proof system given by Dantchev, Martin, and Szeider in [7].

Definition 5 (Dantchev, Martin, Szeider [7])**.** *A* parameterized proof system *for a parameterized language $L \subseteq \Sigma^* \times \mathbb{N}$ is a function $P : \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$ such that $rng(P) = L$ and $P(x, k)$ can be computed in time $O(f(k)|x|^{O(1)})$ with some computable function f.*

The purpose of the second argument in P remains a little bit unclear to us since all natural proof systems we can think of do not have this feature. Thus, we propose the following simplification.

Definition 6. *A* proof system *for a parameterized language $L \subseteq \Sigma^* \times \mathbb{N}$ is a polynomial-time computable function $P : \Sigma^* \to \Sigma^* \times \mathbb{N}$ such that $rng(P) = L$.*

And now we would like to show that both versions are even formally equivalent in the sense that a parameterized language has a proof system in which all strings possess "short" proofs if and only if it has a parameterized proof system with this property. First we have to formalize the notion of "short". In the framework of [7] it goes as follows:

Definition 7 (Dantchev, Martin, Szeider [7])**.** *A parameterized proof system P for a parameterized language L is* fpt-bounded *if there exist computable functions f and g such that every $(x, k) \in L$ has a P-proof (y, k') with $|y| \leq f(k)|x|^{O(1)}$ and $k' \leq g(k)$.*

Again, our analogue is simpler.

Definition 8. *A proof system P for a parameterized language L is* fpt-bounded *if there exists a computable function f such that every $(x, k) \in L$ has a P-proof of size at most $f(k)|x|^{O(1)}$.*

Recall that by the theorem of Cook and Reckhow [6], the class of all languages with polynomially bounded proof systems coincides with NP. To obtain a similar result in the parameterized world, we use the following parameterized version of NP.

Definition 9 (Flum, Grohe [9])**.** *The class* $\mathsf{para}\text{-}\mathsf{NP}$ *contains all parameterized languages which can be decided by a nondeterministic Turing machine in time $f(k)|x|^{O(1)}$ for a computable function f.*

Theorem 10. *Let $L \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized language. Then the following statements are equivalent:*

1. *There exists an fpt-bounded proof system for L.*
2. *There exists an fpt-bounded parameterized proof system for L.*
3. *$L \in \mathsf{para\text{-}NP}$.*

Proof. For the implication $1 \Rightarrow 2$, let P be an fpt-bounded proof system for L. Then the system P' defined by $P'(y,k) = P(y)$ is an fpt-bounded parameterized proof system for L.

For the implication $2 \Rightarrow 3$, let P be an fpt-bounded parameterized proof system for L such that every $(x,k) \in L$ has a P-proof (y,k') with $|y| \le f(k)p(|x|)$ and $k' \le g(k)$ for some computable functions f, g and some polynomial p. Let M be a Turing machine computing P in time $h(k)q(n)$ with computable h and a polynomial q. Then $L \in \mathsf{para\text{-}NP}$ by the following algorithm: on input (x,k) we guess a proof (y,k') with $|y| \le f(k)p(|x|)$ and $k' \le g(k)$. Then we verify that $P(y,k') = (x,k)$ in time $h(k')q(|y|)$ which by the choice of (y,k) yields an fpt running time. If the test is true, then we accept the input (x,k), otherwise we reject.

For the implication $3 \Rightarrow 1$, let $L \in \mathsf{para\text{-}NP}$ and let M be a nondeterministic Turing machine for L running in time $f(k)p(n)$ where f is computable and p is a polynomial. Then we define the following proof system P for L:

$$P(x,k,w) = \begin{cases} (x,k) & \text{if } w \text{ is an accepting computation of } M \text{ on input } (x,k) \\ (x_0,k_0) & \text{otherwise} \end{cases}$$

where $(x_0,k_0) \in L$ is some fixed instance. Apparently, P can be computed in polynomial time. Moreover, P is fpt-bounded as every $(x,k) \in L$ has a P-proof of size $O(f(k)p(|x|))$. □

It should be remarked here that the resulting transformation of an fpt-bounded parameterized proof system into an fpt-bounded proof system for the same language is absolutely constructive.

2.3 Parameterized Versions of Ordinary Proof Systems

A *literal* is a positive or negated propositional variable and a *clause* is a set of literals. The *width* of a clause is the number of its literals. A clause is interpreted as the disjunction of its literals and a set of clauses as the conjunction of the clauses. Hence clause sets correspond to formulas in CNF.

The system of *Parameterized Resolution* was introduced by Dantchev, Martin, and Szeider [7]. Parameterized Resolution is a refutation system for the set PCon of parameterized contradictions (cf. Definition 4). Given a set of clauses F in variables $x_1, \dots, x_n$ with $(F,k) \in \mathrm{PCon}$, a *Parameterized Resolution refutation* of (F,k) is a Resolution refutation of

$$F \cup \{\neg x_{i_1} \vee \cdots \vee \neg x_{i_{k+1}} \mid 1 \le i_1 < \cdots < i_{k+1} \le n\}. \qquad (1)$$

Thus, in Parameterized Resolution we have built-in access to all parameterized clauses of the form $\neg x_{i_1} \vee \cdots \vee \neg x_{i_{k+1}}$. All these clauses are available in the system, but when measuring the size of a derivation we only count those which appear in the derivation. Note that Parameterized Resolution is actually a proof system for PCon in the sense of Definition 6, i. e., verification proceeds in polynomial time. This definition allows the following straightforward generalization.

Definition 11. *Let $P : \Sigma^* \to Con$ be an ordinary proof system for the language Con of all (ordinary) CNF contradictions. We define the* parameterized version $\widehat{P}$ *of P by letting $\widehat{P}(F, k, x) = (F, k)$ whenever $P(x)$ is an arbitrary subset of the set of axioms* (1). *If $P(x)$ does not have this form, $\widehat{P}(F, k, x)$ outputs something trivial.*

The only specific proof system we would like to comment on is tree-like Parameterized Resolution (as it will be needed in Section 4). As explained in [7], a tree-like Parameterized refutation of (F, k) can equivalently be described as a *boolean decision tree.* A boolean decision tree for (F, k) is a binary tree where inner nodes are labeled with variables from F and leafs are labeled with clauses from F or parameterized clauses $\neg x_{i_1} \vee \cdots \vee \neg x_{i_{k+1}}$. Each path in the tree corresponds to a partial assignment where a variable x gets value 0 or 1 according to whether the path branches left or right at the node labeled with x. The condition on the decision tree is that each path α must lead to a clause which is falsified by the assignment corresponding to α. Therefore, a boolean decision tree solves the *search problem* for (F, k) which, given an assignment α, asks for a clause falsified by α. It is easy to verify that each tree-like Parameterized Resolution refutation of (F, k) yields a boolean decision tree for (F, k) and vice versa, where the size of the Resolution proof equals the number of nodes in the decision tree.

An embarrassing fact about Parameterized Proof Complexity (that was brought to our attention by an anonymous referee of a previous version of this paper) is that, as defined in Definition 11, $\widehat{P}$ is *never* bounded for some dull reasons.

Example 1. *Let (F, k) be the parameterized contradiction in which F is the set of positive clauses $\{x_{1,1} \vee \ldots \vee x_{1,n}, \ldots, x_{k+1,1} \vee \ldots \vee x_{k+1,n}\}$. Then in order to make this set even* semantically *invalid, one has to append to it all n^{k+1} parameterized axioms of the form $\neg x_{1,j_1} \vee \ldots \vee \neg x_{k+1,j_{k+1}}$.*

Obviously, this is not the kind of phenomena we want to study (and not the kind of methods we want to develop) so we have to try to somehow isolate such pathological examples. One approach (borrowed from circuit complexity) would be simply to declare some parameterized contradictions "natural", "interesting" or "explicit" without giving precise definitions or even revealing exact reasons for this classification. Another possibility (that we adopt in this paper) is to *formally* restrict the set of contradictions we are interested in.

Definition 12. *A parameterized contradiction (F, k) is* strong *if F itself is a contradiction. A proof system P for the set* PCon *is* weakly fpt-bounded *if there exists a computable function f such that every* **strong** $(F, k) \in$ PCon *has a P-proof of size at most $f(k)|F|^{O(1)}$.*

One reason to introduce this restriction is that "interesting" contradictions are almost always strong. In fact, the only exception we are aware of (even if it is the one that inspired almost all material in Section 4) is the vertex cover problem.

On a more philosophical level, the concept of a strong parameterized contradiction intends to capture the idea that the new knowledge provided by parameterized axioms should be rather thought of as a *helper* or an *additional feature* made available to already existing DPLL algorithms rather than being the prime source of the *validity* of the statement.

Finally, we are not aware of any analogue of Example 1 for strong parameterized contradictions.

Yet another possibility to get rid of this example is to try to encode parameterized axioms in (1) in a more economical way (so that their number stays small), possibly using some auxiliary variables. For Parameterized Resolution this possibility was discussed already in [7], and we continue this discussion in a broader context in Section 5.

3 Parameterized Resolution is Not Weakly fpt-Bounded

The *pigeonhole principle* PHP_n^{n+1} uses variables $x_{i,j}$ with $i \in [n+1]$ and $j \in [n]$, indicating that pigeon i goes into hole j. PHP_n^{n+1} consists of the clauses

$$\bigvee_{j\in[n]} x_{i,j} \quad \text{for all pigeons } i \in [n+1]$$

and $\neg x_{i_1,j} \vee \neg x_{i_2,j}$ for all choices of distinct pigeons $i_1, i_2 \in [n+1]$ and holes $j \in [n]$.

Let F_d be the fragment of the Frege system over de Morgan basis $\{\neg, \wedge, \vee\}$ that operates with formulas of logical depth at most d.

Theorem 13. *For any fixed $d, k \geq 0$ and all sufficiently large n, any refutation of (PHP_n^{n+1}, k) in $\widehat{F_d}$, the parameterized version of F_d, requires size $\geq n^{k/5}$.*

Note that, somewhat surprisingly, d does not appear in the final bound at all (although it implicitly appears in the bound assumed in the "sufficiently large" premise).

Proof. Choose uniformly at random a set I of $n - \sqrt{n}$ pigeons and match them with a set J of $n - \sqrt{n}$ uniformly chosen holes. Such partial matching f induces the following natural partial assignment of the variables of PHP_n^{n+1}:

$$\begin{array}{ll} x_{i,j} = 1 & \text{whenever } i \in I \text{ and } f(i) = j, \\ x_{i,j} = 0 & \text{whenever } i \in I \text{ and } f(i) \neq j, \\ x_{i,j} = 0 & \text{whenever } j \in J \text{ and there exist } i' \neq i \text{ such that } f(i') = j, \text{ and} \\ x_{i,j} = \star & \text{otherwise.} \end{array}$$

We claim that with non-zero probability such partial assignment satisfies all parameterized axioms used in the refutation, as long as there are at most $n^{k/5}$ of them. (Notice that we do not care if such assignment falsifies unused parameterized axioms.) Before proving this claim, we show how the result follows from it.

The restricted refutation will not contain any parameterized axiom. Thus it is a classical F_d-refutation for the restricted formula, which in turn is equivalent (up to a re-indexing of pigeons and holes) to $PHP_{\sqrt{n}}^{\sqrt{n}+1}$. Such refutation must be of size at least $2^{n^{c_d}}$ [13,15] for some $c_d > 0$, thus bigger than $n^{k/5}$ if n is sufficiently large. This concludes the proof.

The missing part is to show that the probabilistic choice of the partial matching realizes the desired properties with positive probability. Consider a parameterized axiom $\neg x_{i_1,j_1} \vee \ldots \vee \neg x_{i_{k+1},j_{k+1}}$. If there are two equal indexes j_a and j_b for $a \neq b$, then such axiom is just a weakening of a standard clause of the pigeonhole principle and does not need any special treatment.

We can now focus on a parameterized axiom in which exactly $k+1$ holes are represented: the probability that such axiom fails to be satisfied is the probability that all x_{i_l,j_l} are either true or unassigned for $1 \leq l \leq k+1$. Let $J_0 = \{j_1, \ldots, j_{k+1}\}$ be the set of all holes represented in our axiom. The probability that the support J of our random restriction contains at most $k/2$ of them (and hence the complement to J that has size $\sqrt{n}$ contains at least $k/2$ of them) is bounded by $\binom{k+1}{k/2} \cdot \frac{\binom{n-k/2}{\sqrt{n}-k/2}}{\binom{n}{\sqrt{n}}} \leq 2^{k+1} n^{-k/4}$. And, conditioned by the event $|J \cap J_0| \geq k/2$, the probability that every hole $j_a \in J \cap J_0$ is sent by the matching f to the right pigeon i_a (so that x_{i_a,j_a} is not set to 0) is at most $(n - k/2)^{-k/2}$. Thus, the overall probability that our random partial assignment does not satisfy an individual parameterized axiom is bounded by $2^{k+1}n^{-k/4} + (n - k/2)^{-k/2} < n^{-k/5}$ for sufficiently large n. By the union bound, if our refutation has size $\leq n^{k/5}$, then for at least one particular choice of f the corresponding assignment satisfies all parameterized axioms actually used in the refutation that, as we already observed, finishes the proof. $\square$

The same proof clearly works for weaker versions of the pigeonhole principle, like functional or onto.

4 Kernels and Small Refutations

The notion of *efficient kernelization* plays an important role in the theory of parameterized complexity. A kernelization for a parameterized language L is a polynomial-time procedure $A : \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$ such that for each (x, k)

1. $(x, k) \in L$ if and only if $A(x, k) \in L$ and
2. if $A(x, k) = (x', k')$, then $k' \leq k$ and $|x'| \leq f(k)$ for some computable function f independent of $|x|$.

It is clear that if a parameterized language admits a kernelization, then the language is fixed-parameter tractable, but also the converse is true (cf. [10]). For parameterized proof complexity we suggest a similar notion of kernel for parameterized contradictions:

Definition 14. *A set $\Gamma \subseteq$ PCon of parameterized contradictions has a* kernel *if there exists a computable function f such that every $(F, k) \in \Gamma$ has a subset $F' \subseteq F$ of clauses satisfying the following conditions:*

1. F' contains at most $f(k)$ variables and

2. (F', k) is a parameterized contradiction.

Clearly, any $k + 1$ positive clauses of width $f(k)$ and with pairwise disjoint sets of variables make a kernel that we will call a *trivial kernel.* It is very easy to come up with many parameterized contradictions (pebbling contradictions, colorability, sparse pigeonhole principle etc.) that possess trivial kernels. The interesting questions here seem to be the following:

1. Do there exist "natural" parameterized contradictions that possess only non-trivial kernels? And do we have a "parameterized automatizability", i.e. , is it easy to find a kernel once we know that it exists?
2. Do there exist "natural" parameterized contradictions that, contrary to the situation in computational complexity, have fpt-bounded refutations despite the fact that they do not have any kernel at all?

In this section we are trying to address both questions. For question 1, our motivating example is the *vertex cover* problem. A vertex cover for a graph G is a set $C \subseteq V(G)$ such that for any $\{u, v\} \in E(G)$ either $u \in C$ or $v \in C$. To determine whether G has a vertex cover of size at most k there is a well-known [8, Chapter 3] fixed parameter tractable algorithm (here the parameter is k). This algorithm is based on the following observation: if a vertex is not in C, then all its neighbors must be in C. The algorithm is a simple recursive procedure which focuses on an arbitrary vertex u, and on its neighbors $v_1, \ldots, v_l$: if neither $G \setminus \{u\}$ has a vertex cover of size $k-1$ nor $G \setminus \{u, v_1, \ldots, v_l\}$ has a vertex cover of size $k - l$, then G has no vertex cover of size k.

This is interpretable as a parameterized DPLL procedure on the 2-CNF $F_G = \bigwedge_{\{u,v\} \in E(G)} (x_u \vee x_v)$ where x_u indicates whether $u \in C$. The DPLL procedure fixes an arbitrary variable x_u and splits on it. When $x_u = 1$, then the DPLL algorithm proceeds with analyzing $F_G \restriction_{x_u=1}$ which is equal to $F_{G \setminus \{u\}}$. When $x_u = 0$, then $x_{v_1} = 1, \ldots, x_{v_l} = 1$ by unit propagation. Thus the DPLL proceeds on formula $F_G \restriction_{\{x_u=0, x_{v_1}=1, \ldots, x_{v_l}=1\}} = F_{G \setminus \{u, v_1, \ldots, v_l\}}$. If at any point the DPLL has more than k variables set to one, it stops and backtracks.

And now we establish a far-reaching generalization of this example.

Theorem 15. *If F is a CNF of width d and (F, k) is a parameterized contradiction, then (F, k) has a tree-like Parameterized Resolution refutation of size $O(d^{k+1})$. Moreover, there is an algorithm that for any (F, k) either finds such tree-like refutation or finds a satisfying assignment for F of weight $\leq k$. The algorithm runs in time $O(|F| \cdot k \cdot d^{k+1})$.*

Proof. Assume (F, k) is a parameterized contradiction. We want to find a refutation for F with parameter k (i.e., at most k variables can be set to true). We first consider a clause $C = x_1 \vee x_2 \vee \ldots \vee x_l$ where $l \leq d$ with all positive literals. Such clause exists because otherwise the full zero assignment would satisfy F.

By induction on k we will prove that (F, k) has a parameterized tree-like refutation of size at most $2 \cdot \sum_{i=0}^{k+1} d^i - 1$. For $k = 0$ the clauses $\{\neg x_i\}_{i=1}^{l}$ are parameterized axioms of the system, thus C is refutable in size at most $1 + 2l \leq 1 + 2d$.

Now consider $k > 0$. For any $1 \leq i \leq l$, let F_i be the restriction of F obtained by setting $x_i = 1$. Each $(F_i, k-1)$ is a parameterized contradiction, otherwise (F, k) would not be. By inductive hypothesis $(F_i, k-1)$ has a tree-like refutation of size at most $s = 2 \sum_{i=0}^{k} d^i - 1$. This refutation can be turned into a tree-like derivation of $\neg x_i$ from (F, k). Now we can derive all $\neg x_i$ for $1 \leq i \leq l$ and refute clause C. Such refutation has length $1 + l + ls \leq 1 + d + ds = 2 \cdot \sum_{i=0}^{k+1} d^i - 1$.

By inspection of the proof, it is clear that the refutation can be computed by a simple procedure which at each step looks for a clause C with only positive literals, and builds a refutation of (F, k) recursively by: building l refutations of $(F_i, k-1)$; turning them in l derivations $(F, k) \vdash \neg x_i$; and resolving against C. This procedure can be easily implemented in the claimed running time.

So far we considered (F, k) to be a parameterized contradiction. If that is not the case, then the algorithm fails. It can fail in two ways: (a) it does not find a clause with only positive literals; (b) one among $(F_i, k-1)$ is not a parameterized contradiction. The algorithm will output the full zero assignment in case (a) and $\{x_i = 1\} \cup \alpha$ in case (b), where α is an assignment witnessing $(F_i, k-1) \notin \mathrm{PCon}$. By induction we can show that on input (F, k) this procedure returns a satisfying assignment of weight $\leq k$. $\square$

We state two interesting consequences of this result.

Corollary 16. *For each $d \in \mathbb{N}$, the set of all parameterized contradictions in d-CNF has a kernel.*

Proof. The refutations constructed in Theorem 15 contain $O(d^k)$ initial clauses in $O(d^{k+1})$ variables. These clauses form a kernel. $\square$

The following corollary expresses some restricted form of automatizability (cf. also the discussion in Section 5).

Corollary 17. *If* $\Gamma \subseteq \mathrm{PCon}$ *has a kernel, then there exists an fpt-algorithm which on input* $(F, k) \in \Gamma$ *returns both a kernel and a refutation of* (F, k)*.*

Proof. Let Γ have a kernel of size $f(k)$. Then the kernel only contains clauses of width $\leq f(k)$. On input (F, k) we run the algorithm of Theorem 15 on the CNF formula consisting of all clauses of F with width $\leq f(k)$. This yields a kernel together with its refutation. □

Let us now turn to our second question. As we already observed in the introduction, a very simple example of a parameterized contradiction having an fpt-bounded refutation despite not having any kernel at all is provided by $(x_1 \vee x_2 \vee \ldots \vee x_n) \wedge \neg x_1 \wedge \neg x_2 \wedge \ldots \wedge \neg x_n$. We conclude this section by presenting a more meaningful one.

The *linear ordering principle LOP* claims that any linearly ordered set has a minimal element. Its propositional formulation can be described as follows:

LOP:

$\neg x_{i,j} \vee \neg x_{j,i}$	for every i, j	(Antisymmetry)
$\neg x_{i,j} \vee \neg x_{j,k} \vee x_{i,k}$	for every i, j, k	(Transitivity)
$\bigvee_{j \in [n] \setminus \{i\}} x_{j,i}$	for every i	(Predecessor)
$x_{i,j} \vee x_{j,i}$	for every i, j	(Totality)

Totality axioms provide a trivial kernel and thus this principle is not good for our purposes. We, however, can modify it as follows.

LOP*: we consider only variables $x_{i,j}$ for $i < j$. The intended meaning is that $x_{i,j}$ is true whenever j precedes i in the ordering, and false if i precedes j. The reader may think $x_{i,j}$ to indicate if i and j are an inversion in the permutation for the indexes described by the total order. In particular the full true assignment represents the linear order $(n, n-1, n-2, \ldots, 2, 1)$ while the full false assignment represents $(1, 2, \ldots, n-2, n-1, n)$. This representation will help in the proof of Theorem 18.

LOP^*_n is obtained by substituting in LOP_n any occurrence of $x_{j,i}$ for $j > i$ with $\neg x_{i,j}$. In this way all totality and antisymmetry clauses vanish, and transitivity translates according to relative ranks of the involved indexes.

$\neg x_{i,j} \vee \neg x_{j,k} \vee x_{i,k}$	for all $i < j < k$	(Transitivity 1)
$x_{i,j} \vee x_{j,k} \vee \neg x_{i,k}$	for all $i < j < k$	(Transitivity 2)
$\bigvee_{j<i} \neg x_{j,i} \vee \bigvee_{i<j} x_{i,j}$	for all i	(Predecessor)

The alternative formulation LOP^* does not have a kernel because all clauses of bounded width are satisfiable by the all zero assignment which represents a total order. Nevertheless LOP^* admits fpt-bounded tree-like refutations.

Theorem 18. LOP^*_n *has fpt-bounded tree-like refutations in Parameterized Resolution.*

Proof. Let (LOP^*_n, k) be the given instance and assume w. l. o. g. that $k \leq n$. We are going to derive LOP^*_{k+1} from LOP^*_n in polynomial length. This concludes the proof of the theorem because LOP^*_{k+1} has $O(k^2)$ variables and consequently has a tree-like refutation of length $2^{O(k^2)}$.

The idea of the refutation is that for any total order either the least element is among $1, \ldots, k+1$ or there is an element less than all of them. This means that there are at least $k+1$ inversions with respect to the canonical order (i. e., $k+1$ variables are set to 1). To obtain LOP^*_{k+1} we have to derive

$$\bigvee_{1 \leq j < i} \neg x_{j,i} \vee \bigvee_{i < j \leq k+1} x_{i,j}$$

for any $1 \leq i \leq k+1$. W. l. o. g. we discuss the case $i = 1$ which requires simpler notation, the other k cases are analogous.

Our goal then is to derive $\bigvee_{1<j\leq k+1} x_{1,j}$. For any $l > k+1$ consider the following clauses: the first is an axiom of Parameterized Resolution, the others are transitivity axioms.

$$\neg x_{1,l} \vee \neg x_{2,l} \vee \ldots \vee \neg x_{k+1,l} \tag{2}$$

$$x_{1,2} \vee x_{2,l} \vee \neg x_{1,l} \tag{3}$$

$$x_{1,3} \vee x_{3,l} \vee \neg x_{1,l} \tag{4}$$

$$\vdots$$

$$x_{1,k+1} \vee x_{k+1,l} \vee \neg x_{1,l} \tag{5}$$

By applying Resolution between clause (2) and the transitivity clauses we obtain

$$x_{1,2} \vee x_{1,3} \vee \ldots \vee x_{1,k+1} \vee \neg x_{1,l} \tag{6}$$

We just proved that if 1 is the least index among the first $k+1$, then no index above $k+1$ can be less than 1, otherwise there would be at least $k+1$ true variables. The predecessor constraint for 1 contains the literal $x_{1,l}$ for every l; thus applying Resolution between that and clause (6) for every $l > k+1$ yields $\bigvee_{1<j\leq k+1} x_{1,j}$.

We obtained the predecessor axiom for index 1 in LOP^*_{k+1} by a derivation of size $O(kn)$. With $k+1$ such deductions we obtain LOP^*_{k+1}. As the whole refutation of LOP^*_n has length $O(k^2 n) + 2^{O(k^2)}$, it is fpt-bounded. □

5 Discussion and Open Problems

Our lower bound technique is very general and works for any tautology that is not killed by a "large" restriction. Are there any other techniques to prove the hardness of parameterized contradictions? One concrete question that we may ask along these lines is this:

Question 1. *Does (PHP^{2n}_n, k) have an fpt-bounded refutation in Parameterized Resolution?*

Is the parameterized proof system $\widehat{P}$ from Definition 11 the most natural way to define the parameterized analogue of P? The answer depends on the original proof system P, of course. The main (unspoken) reason why [7] defined it in this way is simply because weak proof systems cannot directly talk about the weight of the input. Let us first discuss two familiar systems that are strong enough to overcome this limitation: Frege and Cutting Planes.

The problem of getting super-polynomial lower bounds for the Frege proof system F is one of the biggest open problems in Logic and Theoretical Computer Science. The question whether its parameterized version $\widehat{F}$ is *weakly* fpt-bounded is even harder (as we just add new axioms). A similar conclusion remains true if we combine all parameterized axioms into one (using e.g. [5]) but allow arbitrary parameterized contradictions, not necessarily strong. There is not much more we can add here.

The case of Cutting Planes (CP) is way more interesting. First of all, we do not seem to know lower bounds even for the "canonical" version $\widehat{CP}$:

Question 2. *Is $\widehat{CP}$ weakly fpt-bounded?*

This, of course, is yet another reflection of the mysterious status of this proof system: the only known lower bounds for it are based on very indirect methods (interpolation, see [4, 16]) and no direct, combinatorial proof is currently known. And if we try to generalize the methods from [4, 16] (at least in a straightforward way) then we immediately arrive at a problem in parameterized circuit complexity that seems to be widely open (at least, we do not see how known methods can be applied to it).

Question 3. *Find an explicit* partial *monotone function (a.k.a. a monotone promise problem) $f : [n]^{\leq k} \to \{0,1\}$ defined only on inputs of Hamming weight $\leq k$ that does not possess monotone circuits of size $f(k)n^{O(1)}$.*

Note that the problem of finding (say) a $\sqrt{k}$-clique does become easy in this context.

For weaker proof systems, [7, Section 4] proposed to use auxiliary variables. Their suggestion was to add new "pigeonhole variables" $p_{i,j}$ $(i \in [n],\ j \in [k])$ and "pigeonhole clauses"

$$\neg x_i \vee \bigvee_{j\in[k]} p_{i,j} \qquad \text{for all } i \in [n] \qquad \text{(Pigeon clauses)}$$

$$\neg p_{i_1,j} \vee \neg p_{i_2,j} \qquad \text{for all } i_1 \neq i_2 \in [n],\ j \in [k], \qquad \text{(Hole clauses)}$$

where $x_1, \ldots, x_n$ are the original variables. Remarkably, they proved that the pigeonhole principle becomes fpt-bounded in this version of Parameterized Resolution.

Also, the disturbing Example 1 turns into an instance of PHP_k^{k+1} with large "metapigeons" that does have an fpt-bounded proof (e.g., the straightforward adaption of the rectangular proof from [17, Example 1]). Thus, following [7], we ask:

Question 4. *Is Parameterized Resolution with auxiliary variables fpt-bounded?*

Let us now point out that there is an interesting and well-studied class of contradictions for which the difference between these two encodings disappears, and these are independent set principles. Following [2], let G be a graph $[n]$ in which vertices are split into k subsets $V_1, \ldots, V_k$ of size n/k each called *blocks*. The principle $\alpha_{block}(G, k)$ encodes the fact that G has a block-respecting independent set of size k; it has the variables x_v $(v \in [n])$ and the axioms

$$\neg x_u \vee \neg x_v \qquad \text{for all } (u, v) \in E(G) \qquad \text{(Edge clauses)}$$

$$\bigvee_{v\in V_i} x_v \qquad \text{for all } i \in [k] \qquad \text{(Block clauses)}$$

$$(\neg x_u \vee \neg x_v) \qquad \text{for all } u \neq v \text{ in the same block} \qquad \text{(1–1 clauses).}$$

The fact that all satisfying assignments have at most k ones is already built in this principle: all parameterized axioms are subsumed by the 1–1 clauses above. Auxiliary clauses in the sense of [7] (both pigeon and holes) also do not help to reduce the refutation size, as witnessed by the following substitution of the pigeonhole variables:

$$p_{v,j} \mapsto \begin{cases} 0 & \text{if } v \notin V_j \\ x_v & \text{if } v \in V_j. \end{cases}$$

Thus, we are also asking the following specific form of Question 4:

Question 5. *Do the principles $\alpha_{block}(G, k)$ always have fpt-bounded refutations as long as the graph G does not contain block-respecting independent sets of size k?*

One good candidate for a lower bound here would be Erdös-Rényi random graphs $G(n, p)$ for an appropriately chosen value of p.

Let us recall that a proof system P is *automatizable* if there exists an algorithm which for a tautology F with a P-proof of size S finds a P-proof for F of size at most $S^{O(1)}$ and runs in time $S^{O(1)}$. Alekhnovich and Razborov [1] proved that if (classical) Resolution or tree-like Resolution were automatizable, then $\mathsf{W[P]}$ coincides with FPR, the randomized version of FPT. On the other hand, tree-like Resolution is quasi-polynomially automatizable (see e.g. [3]).

We point out that the concept of quasi-polynomial automatizability is meaningless in the context of Parameterized Resolution, because every $(F, k) \in \text{PCon}$ with $|F| = n$ has a refutation of size $c \cdot \binom{n}{k+1}$ for some constant c. If $k \le \log n$ this is smaller than $n^{\log n}$; otherwise $\binom{n}{k+1} \le 2^{(k+1)^2}$ which is fpt with respect to k.

On the contrary, the concept of polynomial automatizability can be extended to parameterized proof systems in an obvious way. Thus, we ask:

Question 6. *Is (tree-like) Parameterized Resolution, with or without auxiliary variables, fpt-automatizable or fpt-automatizable w.r.t. strong contradictions? That is, does there exist an algorithm that for any (strong) parameterized contradiction $(F, k) \in$* PCon *outputs its refutation within time $f(k)S^{O(1)}$, where S is the minimal possible size of a parameterized refutation of (F, k)?*

Naturally, unconditional results of this sort are completely out of reach for the moment, so we are willing to allow here any reasonable complexity assumption (that will most likely reside in the realm of parameterized complexity itself).

References

[1] M. Alekhnovich and A. A. Razborov. Resolution is not automatizable unless W[P] is tractable. *SIAM Journal on Computing*, 38(4):1347–1363, 2008.

[2] P. Beame, R. Impagliazzo, and A. Sabharwal. The resolution complexity of independent sets and vertex covers in random graphs. *Computational Complexity*, 16(3):245–297, 2007.

[3] P. Beame, R. M. Karp, T. Pitassi, and M. E. Saks. The efficiency of resolution and Davis–Putnam procedures. *SIAM J. Comput.*, 31(4):1048–1075, 2002.

[4] M. L. Bonet, T. Pitassi, and R. Raz. Lower bounds for cutting planes proofs with small coefficients. *The Journal of Symbolic Logic*, 62(3):708–728, 1997.

[5] S. R. Buss. Polynomial size proofs of the propositional pigeonhole principle. *The Journal of Symbolic Logic*, 52:916–927, 1987.

[6] S. A. Cook and R. A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979.

[7] S. S. Dantchev, B. Martin, and S. Szeider. Parameterized proof complexity. In *Proc. 48th IEEE Symposium on the Foundations of Computer Science*, pages 150–160, 2007.

[8] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer-Verlag, Berlin Heidelberg, 1999.

[9] J. Flum and M. Grohe. Describing parameterized complexity classes. *Information and Computation*, 187(2):291–319, 2003.

[10] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer-Verlag, Berlin Heidelberg, 2006.

[11] Y. Gao. Data reductions, fixed parameter tractability, and random weighted d-CNF satisfiability. *Artificial Intelligence*, 173(14):1343–1366, 2009.

[12] A. Haken. The intractability of resolution. *Theor. Comput. Sci.*, 39:297–308, 1985.

[13] J. Krajíček, P. Pudlák, and A. Woods. Exponential lower bounds to the size of bounded depth Frege proofs of the pigeonhole principle. *Random Structures and Algorithms*, 7(1):15–39, 1995.

[14] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2006.

[15] T. Pitassi, P. Beame, and R. Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Computational Complexity*, 3:97–140, 1993.

[16] P. Pudlák. Lower bounds for resolution and cutting planes proofs and monotone computations. *The Journal of Symbolic Logic*, 62(3):981–998, 1997.

[17] A. Razborov, A. Wigderson, and A. Yao. Read-once branching programs, rectangular proofs of the pigeonhole principle and the transversal calculus. *Combinatorica*, 22(4):555–574, 2002.

[18] S. Riis. A complexity gap for tree resolution. *Computational Complexity*, 10(3):179–209, 2001.