

Parameterized Bounded-Depth Frege Is not Optimal

OLAF BEYERSDORFF, University of Leeds
 NICOLA GALESÌ and MASSIMO LAURIA, Sapienza University Rome
 ALEXANDER A. RAZBOROV, University of Chicago

A general framework for parameterized proof complexity was introduced by Dantchev et al. [2007]. There, the authors show important results on tree-like Parameterized Resolution—a parameterized version of classical Resolution—and their gap complexity theorem implies lower bounds for that system.

The main result of this article significantly improves upon this by showing optimal lower bounds for a parameterized version of bounded-depth Frege. More precisely, we prove that the pigeonhole principle requires proofs of size $n^{\Omega(k)}$ in parameterized bounded-depth Frege, and, as a special case, in dag-like Parameterized Resolution. This answers an open question posed in Dantchev et al. [2007]. In the opposite direction, we interpret a well-known technique for FPT algorithms 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 CNFs.

Categories and Subject Descriptors: F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems—Complexity of proof procedures

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Proof complexity, parameterized complexity, Resolution, bounded-depth Frege

ACM Reference Format:

Beyersdorff, O., Galesi, N., Lauria, M., and Razborov, A. A. 2012. Parameterized bounded-depth frege is not optimal. *ACM Trans. Comput. Theory* 4, 3, Article 7 (September 2012), 16 pages.
 DOI = 10.1145/2355580.2355582 <http://doi.acm.org/10.1145/2355580.2355582>

1. INTRODUCTION

Recently, Dantchev et al. [2007] introduced the framework of *parameterized proof complexity*, an extension of the proof complexity approach of Cook and Reckhow [1979] to parameterized complexity. One motivation for this is the quest for efficient algorithms

This research was supported by a grant from the John Templeton Foundation and a DAAD grant. M. Lauria was partially supported by the Eduard Čech Center for Algebra and Geometry (Prague). A. A. Razborov was partially supported by the Russian Foundation for Basic Research, and at Toyota Technological Institute, Chicago.

Part of this research was done while O. Beyersdorff was visiting Sapienza University Rome. Part of this work was done while A. A. Razborov was at Steklov Mathematical Institute.

A preliminary version of this article appeared in *Proceedings of the 38th International Colloquium on Automata, Languages, and Programming (ICALP'11)* [Beyersdorff et al. 2011b].

Authors' addresses: O. Beyersdorff, School of Computing, University of Leeds, Leeds LS2 9JT, UK; email: o.beyersdorff@leeds.ac.uk; N. Galesi and M. Lauria, Dipartimento di Informatica, Università degli Studi di Roma "La Sapienza", Piazzale Aldo Moro 5, 00185 Roma, Italy; email: {galesi, lauria}@di.uniroma1.it; A. A. Razborov, Department of Computer Science, The University of Chicago, 5801 South Ellis Avenue, Chicago, IL 60637; email: razborov@cs.uchicago.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permission may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701, USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2012 ACM 1942-3454/2012/09-ART7 \$15.00

DOI 10.1145/2355580.2355582 <http://doi.acm.org/10.1145/2355580.2355582>

solving fragments of classically hard problems [Downey and Fellows 1999; Flum and Grohe 2003; Niedermeier 2006]. 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 [2009] 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 different way than in the classical approach: we say that a problem is *fixed-parameter tractable* (FPT) with parameter k if any instance of size n 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. Consider, for example, the classical satisfiability problem of finding a truth assignment that satisfies all clauses of a formula in conjunctive normal form. BOUNDED CNF SAT and WEIGHTED CNF SAT are parameterized variants of CNF satisfiability in which the satisfying assignment is required to have Hamming weight at most k or exactly k , respectively. Many parameterized combinatorial problems can be naturally encoded in BOUNDED CNF SAT or 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 all three problems is reflected by their 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, such as FPT and W[P] (see Downey and Fellows [1999], Flum and Grohe [2006], and Niedermeier [2006] for a comprehensive treatment of the field).

Dantchev et al. [2007] laid the foundations to study complexity of proofs in a parameterized setting. The complementary problem of BOUNDED CNF SAT is that of deciding *parameterized contradictions* PCon: it consists of all pairs (F, k) where F is a propositional formula F which has no satisfying assignment of weight $\leq k$. After considering this notion of propositional *parameterized tautologies*, Dantchev et al. [2007] introduced the concepts of parameterized proof systems and of fpt-bounded proof systems (see Section 2 for a discussion). The main motivation behind the work of Dantchev et al. [2007] was that of generalizing the classical approach of Cook and Reckhow [1979] to the parameterized case and working towards a separation of parameterized complexity classes as FPT and W[2] by techniques developed in proof complexity. In fact, we obtain an analogous result to the well-known Cook-Reckhow theorem from Cook and Reckhow [1979]: a parameterized language L has an fpt-bounded proof system if and only if $L \in \text{para-NP}$ (Theorem 2.8).

In Dantchev et al. [2007], (tree-like) *Parameterized Resolution* was defined as a refutation system for the set of parameterized contradictions. If $(F, k) \in \text{PCon}$ is defined on variables x_1, \dots, x_n then 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.

Dantchev et al. [2007] prove an extension of Riis' *gap theorem* [Riis 2001] and obtain a model theoretic classification for the complexity of tree-like Parameterized Resolution refutations for parameterized contradictions originating as propositional encodings of first-order formulas. In particular, their main result implies that tree-like Parameterized Resolution is not fpt-bounded. A similar question for dag-like Parameterized Resolution was left open in Dantchev et al. [2007]. 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^{n+1} requires proofs of size $n^{\Omega(k)}$ not only in Parameterized Resolution but in the much stronger system of bounded-depth Frege. Our result is in sharp contrast with Dantchev et al. [2007, Proposition 17] that gives efficient proofs of PHP_n^{n+1} in Parameterized Resolution using a more sophisticated encoding with auxiliary variables. 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 *kernelization* plays an important role in the theory of parameterized complexity to design fpt-algorithms. Here we propose a notion of *core* for parameterized proof complexity: the core of a parameterized contradiction (F, k) is 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 core, then it can be efficiently refuted in tree-like Parameterized Resolution with a refutation of size independent of the size of F . 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 core of a formula is not explicit or immediate to find. In Theorem 4.2 we prove that contradictions of bounded width have a core and thus very efficient tree-like Parameterized Resolution refutations.

Is the existence of a core 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 $(x_1 \vee x_2 \vee \dots \vee x_n) \wedge \neg x_1 \wedge \dots \wedge \neg x_n$, but this is a bit of a cheating since the CNF itself has a poly-size tree-like refutation. We include a much more interesting example (Proposition 4.5) of a parameterized contradiction, a version of the linear ordering principle, that has fpt-refutations in tree-like Parameterized Resolution without having a core.

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 [1985] in his seminal paper. 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 [2009] suggested to use a standard DPLL algorithm to find refutations of certain random parameterized d -CNFs. Here we prove that bounded width CNFs have a core and hence are efficiently refutable in tree-like Parameterized Resolution (Theorem 4.2). 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 Article

The remaining part of the article 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 Parameterized Bounded-depth Frege has no fpt-bounded refutations for the pigeonhole principle. Section 4 concentrates on upper bounds: we introduce the notion of a core 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 core. 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 $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. 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

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

The classes of the weft hierarchy are usually defined as the closure of a canonical problem under fpt-reductions. For 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 exactly k , where the weight of an assignment α , denoted as $w(\alpha)$, 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 the W[2]-complete problem BOUNDED CNF SAT (cf. Dantchev et al. [2007]). We include the full proof for this claim.

PROPOSITION 2.1 [DANTCHEV ET AL. 2007]. BOUNDED CNF SAT is W[2]-complete.

PROOF. We first prove that BOUNDED CNF SAT is in W[2] and then that it is W[2]-hard. The first statement is obvious: consider a CNF $F(x_1, \dots, x_n)$ and a parameter k . Introduce new (dummy) variables y_1, \dots, y_k . Then the formula $F(x_1, \dots, x_n, y_1, \dots, y_k)$ has a satisfying assignment of weight k if and only if $F(x_1, \dots, x_n)$ has a satisfying assignment of weight at most k . This yields an fpt-reduction from BOUNDED CNF SAT to WEIGHTED CNF SAT, which implies that BOUNDED CNF SAT is in W[2].

W[2]-hardness of BOUNDED CNF SAT is shown via the converse reduction. Let (F, k) be the input, where F is a CNF in variables x_1, \dots, x_n . Consider the following CNF ψ which in addition to x_1, \dots, x_n uses new variables $y_{i,j}$ for $i \in [n]$ and $j \in [k]$:

$$\begin{array}{ll} \bigvee_i y_{i,j} & \text{for any } j \in [k] \\ \neg y_{i,j} \vee \neg y_{i',j} & \text{for any } i \neq i' \in [n] \text{ and } j \in [k] \\ \neg y_{i,j} \vee \neg y_{i,j} & \text{for any } i \in [n] \text{ and } j \neq j' \in [k] \\ y_{i,1} \vee y_{i,2} \vee \dots \vee y_{i,k} \vee \neg x_i & \text{for any } i \in [n] \\ \neg y_{i,j} \vee x_i & \text{for any } i \in [n] \text{ and } j \in [k]. \end{array}$$

Formula ψ is satisfiable if and only if there is a set of k indices in $[n]$ matched with $[k]$. Variable x_i is true if and only if i is in this set. Thus, any satisfying assignment for ψ has weight $2k$. The fpt-reduction from WEIGHTED CNF SAT to BOUNDED CNF SAT is given by $(F, k) \mapsto (F \wedge \psi, 2k)$. This proves that BOUNDED CNF SAT is W[2]-hard. \square

Like in the classical duality between tautologies and satisfiability, the complement of BOUNDED CNF SAT is a complete problem for coW[2]:

Definition 2.2 [Dantchev et al. 2007]. 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 monographs [Downey and Fellows 1999; Flum and Grohe 2006; Niedermeier 2006].

2.2. Parameterized Proof Systems

We start with discussing the general definition of a parameterized proof system given by Dantchev et al. [2007].

Definition 2.3 [Dantchev et al. 2007]. A *parameterized proof system* for a parameterized language $L \subseteq \Sigma^* \times \mathbb{N}$ is a function $P : \Sigma^* \times \mathbb{N} \rightarrow \Sigma^* \times \mathbb{N}$ such that $\text{rng}(P) = L$ and $P(x, k)$ can be computed in time $f(k)|x|^{O(1)}$ 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 2.4. A *proof system* for a parameterized language $L \subseteq \Sigma^* \times \mathbb{N}$ is a polynomial-time computable function $P : \Sigma^* \rightarrow \Sigma^* \times \mathbb{N}$ such that $\text{rng}(P) = L$.

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 Dantchev et al. [2007] it goes as follows.

Definition 2.5 [Dantchev et al. 2007]. 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 2.6. 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 [1979], 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 2.7 [Flum and Grohe 2003]. The class para-NP contains all parameterized languages that can be decided by a nondeterministic Turing machine in time $f(k)|x|^{O(1)}$ for a computable function f .

THEOREM 2.8. *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) *$L \in \text{para-NP}$.*

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 $|y| \leq f(k)p(|x|)$ and $k' \leq g(k)$ 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 $L \in \text{para-NP}$ by the following algorithm: on input (x, k) we guess a proof (y, k') with $|y| \leq f(k)p(|x|)$ and $k' \leq g(k)$. 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 $L \in \text{para-NP}$ 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. Clearly, P can be computed in polynomial time. Moreover, P is fpt-bounded as every $(x, k) \in L$ has a P -proof of size $O(f(k)p(|x|))$. \square

We remark that the resulting transformation of an fpt-bounded parameterized proof system into an fpt-bounded proof system for the same language is constructive.

2.3. Parameterized Versions of Ordinary Proof Systems

A *literal* is a propositional variable or a negated variable; 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 et al. [2007]. Parameterized Resolution is a refutation system for the set PCon of parameterized contradictions (cf. Definition 2.2). Given a set of clauses F in variables x_1, \dots, x_n with $(F, k) \in \text{PCon}$, a *Parameterized Resolution refutation* of (F, k) is a Resolution refutation of

$$F \cup \{\neg x_{i_1} \vee \dots \vee \neg x_{i_{k+1}} \mid 1 \leq i_1 < \dots < i_{k+1} \leq n\}. \quad (1)$$

Thus, in Parameterized Resolution, we have built-in access to all clauses of the form $\neg x_{i_1} \vee \dots \vee \neg x_{i_{k+1}}$. We call these clauses *parameterized axioms*. All parameterized axioms 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 2.4, that is, verification proceeds in polynomial time. This definition allows the following straightforward generalization.

Definition 2.9. Let $P : \Sigma^* \rightarrow \text{Con}$ be an ordinary proof system for the language Con of all (ordinary) CNF contradictions. We define the *parameterized version* \tilde{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 Dantchev et al. [2007], 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 $\neg x_{i_1} \vee \dots \vee \neg x_{i_{k+1}}$. 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 (brought to our attention by an anonymous referee of a previous version of this article) is that, as defined in Definition 2.9, \widehat{P} is *never* bounded for some dull reasons.

Example 2.10. Let (F, k) be the parameterized contradiction in which F is the set of positive clauses $\{x_{1,1} \vee \dots \vee x_{1,n}, \dots, x_{k+1,1} \vee \dots \vee x_{k+1,n}\}$. 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 $\neg x_{1,j_1} \vee \dots \vee \neg x_{k+1,j_{k+1}}$.

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 article) is to *formally* restrict the set of contradictions we are interested in.

Definition 2.11. 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* $(F, k) \in \text{PCon}$ has a P -proof of size at most $f(k)|F|^{O(1)}$.

One reason to introduce this restriction is that many “interesting” contradictions are 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 2.10 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 Dantchev et al. [2007], and we continue this discussion in a broader context in Section 5.

3. PARAMETERIZED BOUNDED-DEPTH FREGE 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 two distinct pigeons $i_1, i_2 \in [n+1]$ and a hole $j \in [n]$.

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 3.1. *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 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} :

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

We claim that with nonzero 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 theorem follows.

The refutation, restricted with such assignment, does not contain parameterized axioms anymore. 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 $2^{n^{c_d}}$ [Krajíček et al. 1995; Pitassi et al. 1993] 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 $\neg x_{i_1, j_1} \vee \dots \vee \neg x_{i_{k+1}, j_{k+1}}$. 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 $J_0 = \{j_1, \dots, j_{k+1}\}$ 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

$$2^{k+1}n^{-k/4} + (n - k/2)^{-k/2} < n^{-k/5} \quad (2)$$

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. As we already observed, this concludes the proof. \square

The same proof works for weaker versions of the pigeonhole principle, like functional or onto, and it works for nonconstant depth d up to $\Omega(\log \log n)$ (cf. Pitassi et al. [1993, Corollary 11]). If we consider Parameterized Resolution instead of parameterized bounded-depth Frege, our proof applies also to the pigeonhole principle with arbitrarily many pigeons.

THEOREM 3.2. *For any fixed $k \geq 0$ and all sufficiently large n and any $m > n$, any parameterized Resolution refutation of (PHP_n^m, k) requires size $\geq n^{k/5}$.*

PROOF. As in the proof of Theorem 3.1, we uniformly choose at random a set I of $n - \sqrt{n}$ pigeons and match them with a set J of $n - \sqrt{n}$ uniformly chosen holes. This partial matching is naturally associated with a partial assignment that restricts the formula PHP_n^m to a formula $PHP_{\sqrt{n}}^{m-n+\sqrt{n}}$. This formula requires Resolution refutations of size $2^{n^{\Omega(1)}}$ [Raz 2004; Razborov 2004], which is asymptotically bigger than $n^{k/5}$.

The rest of the proof is identical to that of Theorem 3.1, with the only difference that now the bound (2) becomes $2^{k+1}n^{-k/4} + (m - k/2)^{-k/2}$. But this is less than $2^{k+1}n^{-k/4} + (n - k/2)^{-k/2}$ as $m > n$. \square

It is remarkable that this lower bound does not depend on the number of pigeons. This contrasts with classical Resolution, where we have the following anti-monotonicity: for $n + 1$ pigeons there is a lower bound of $2^{\Omega(n)}$, as shown in Haken [1985], while for the case of infinitely many pigeons¹ the smallest Resolution refutation has length between $2^{\Omega(\sqrt[3]{n})}$ and $2^{O(\sqrt{n \log n})}$ (see Buss and Pitassi [1997], Razborov et al. [2002], Raz [2004], and Razborov [2004]).

4. CORES AND SMALL REFUTATIONS

The notion of *kernelization* plays an important role in the theory of parameterized complexity. A kernelization for a parameterized language L is a polynomial-time procedure $A : \Sigma^* \times \mathbb{N} \rightarrow \Sigma^* \times \mathbb{N}$ 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 it is fixed-parameter tractable. The converse is also true for decidable languages (cf. Flum and Grohe [2006]). For parameterized proof complexity, we suggest a similar notion of core for parameterized contradictions.

¹This formula has infinite length, but its smallest refutation has finite length since it uses a finite number of initial clauses. Notice that any PHP_n^m refutation of length L mentions at most L pigeons, thus the Resolution complexity of PHP_n^∞ is equal to that of PHP_n^m for some value of m .

Definition 4.1. A *core* for a set $\Gamma \subseteq \text{PCon}$ of parameterized contradictions is a mapping which maps every $(F, k) \in \Gamma$ to 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,

where f is a computable function depending only on the mapping.

Note that we do not impose any a priori restrictions on the complexity of the mapping itself.

We will sometimes abuse terminology by saying that a set of clauses $F' \subseteq F$ is a core of F when it is clear from the context that F is a member of a family of parameterized contradictions and that F' can be chosen for any k and any F in that family.

Clearly, any $k + 1$ positive clauses of width $f(k)$ and with pairwise disjoint sets of variables make a core that we will call a *trivial core* (cf. Example 2.10). It is very easy to come up with many parameterized contradictions (pebbling contradictions, colorability, sparse pigeonhole principle etc.) that possess trivial cores.

The interesting questions here seem to be the following.

- (1) Do there exist “natural” parameterized contradictions that possess only non-trivial cores? And do we have a “parameterized automatizability”, that is, is it easy to find a core 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 core at all?

In the following, 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 $v \in C$ or both. To determine whether G has a vertex cover of size at most k there is a well-known [Downey and Fellows 1999, 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, \dots, v_l : if neither $G \setminus \{u\}$ has a vertex cover of size $k - 1$ nor $G \setminus \{u, v_1, \dots, v_l\}$ 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 branches on it. When $x_u = 1$, then the DPLL algorithm proceeds with analyzing $F_G \upharpoonright_{x_u=1}$ which is equal to $F_{G \setminus \{u\}}$. When $x_u = 0$, then $x_{v_1} = 1, \dots, x_{v_l} = 1$ by unit propagation. Thus, the DPLL proceeds on formula $F_G \upharpoonright_{\{x_u=0, x_{v_1}=1, \dots, x_{v_l}=1\}} = F_{G \setminus \{u, v_1, \dots, v_l\}}$. 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 4.2. *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 $C = x_1 \vee x_2 \vee \dots \vee x_l$ 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) (by appending this literal to every clause in the derivation). 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 of $(F_i, k - 1)$ 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 $(F_i, k - 1) \notin \text{PCon}$. By induction, we can show that on input (F, k) this procedure returns a satisfying assignment of weight $\leq k$. \square

A related result was obtained by Chen and Flum [2008, Theorem 12]. Notice that while BOUNDED CNF SAT and WEIGHTED CNF SAT are both $W[2]$ -complete, BOUNDED d -CNF SAT is in FPT and WEIGHTED d -CNF SAT is known to be $W[1]$ -complete. This means that reducing the case of exact weight to bounded weight (see the proof of Proposition 2.1) requires large clauses unless $\text{FPT} = W[1]$. We state two interesting consequences of Theorem 4.2.

COROLLARY 4.3. *For each $d \in \mathbb{N}$, the set of all parameterized contradictions in d -CNF has a core.*

PROOF. The refutations constructed in Theorem 4.2 contain $O(d^k)$ initial clauses in $O(d^{k+1})$ variables. These clauses form a core. \square

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

COROLLARY 4.4. *If $\Gamma \subseteq \text{PCon}$ has a core, then there exists an fpt-algorithm that on input $(F, k) \in \Gamma$ returns both a core and a refutation of (F, k) .*

PROOF. Let Γ have a core of size $f(k)$. Then the core only contains clauses of width $\leq f(k)$. On input (F, k) , we run the algorithm of Theorem 4.2 on the CNF formula consisting of all clauses of F with width $\leq f(k)$. This yields a core together with its refutation. \square

4.1. The Linear Ordering Principle

Let us now turn to Question (2), that is whether the existence of a core is 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 $(x_1 \vee x_2 \vee \dots \vee x_n) \wedge \neg x_1 \wedge \dots \wedge \neg x_n$. A more interesting example—a version of the linear ordering principle—is discussed here.

The *linear ordering principle* LOP claims that any linearly ordered set has a minimal element. Its propositional formulation can be described as follows. It has the variables $(x_{i,j} \mid i, j \in [n], i \neq j)$ with the intended meaning that $x_{i,j}$ is false if i precedes j in the ordering and is true if j precedes i . The axioms are given by:

LOP:

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

Totality axioms provide a trivial core: take two disjoint sets $A, B \subseteq [n]$, $|A| = |B| = k+1$. Then the totality axioms $x_{i,j} \vee x_{j,i}$ for $i \in A, j \in B$ form a core over $2k+2$ variables. Thus, this principle is not good for our purposes.

We, however, can modify it to the following version LOP^* by restricting to the variables $x_{i,j}$ for $i < j$. In particular, the full true assignment represents the linear order $(n, n-1, n-2, \dots, 2, 1)$ while the full false assignment represents $(1, 2, \dots, n-2, n-1, n)$. This representation will help in the proof of Proposition 4.5.

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. Thus, we obtain the clauses

$$\begin{array}{lll}
\neg x_{i,j} \vee \neg x_{j,k} \vee x_{i,k} & \text{for all } i < j < k & \text{(Transitivity 1)} \\
x_{i,j} \vee x_{j,k} \vee \neg x_{i,k} & \text{for all } i < j < k & \text{(Transitivity 2)} \\
\bigvee_{j < i} \neg x_{j,i} \vee \bigvee_{i < j} x_{i,j} & \text{for all } i & \text{(Predecessor)}
\end{array}$$

The alternative formulation LOP^* does not have a core because all clauses of bounded width are satisfiable by the all zero assignment which represents a total order. Also, neither LOP_n nor LOP_n^* have a poly-size tree-like resolution refutation (that can be seen, e.g., by inspecting the proof of Bonnet and Galesi [2001] that established this in the absence of totality axioms). Nevertheless, LOP^* admits fpt-bounded tree-like refutations.

PROPOSITION 4.5. LOP_n^* has fpt-bounded tree-like refutations in Parameterized Resolution.

PROOF. The idea of the refutation is that for any total order either the least element is among $1, \dots, k+1$ or there is an element less than all of them. In the latter case, there are at least $k+1$ inversions with respect to the canonical order (i.e., $k+1$ variables are set to 1). In the language of decision trees the proof boils down to querying all the variables which deal with the first $k+1$ elements, and then to compare any other element with the smallest among the first $k+1$.

Let (LOP_n^*, k) be the given instance and assume, without loss of generality, that $k \leq n$. We are going to derive LOP_{k+1}^* from (LOP_n^*, k) 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 $2^{O(k^2)}$.

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 $1 \leq i \leq k+1$. Without loss of generality, 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.

$$\neg x_{1,l} \vee \neg x_{2,l} \vee \dots \vee \neg x_{k+1,l} \quad (3)$$

$$x_{1,2} \vee x_{2,l} \vee \neg x_{1,l} \quad (4)$$

$$x_{1,3} \vee x_{3,l} \vee \neg x_{1,l} \quad (5)$$

⋮

$$x_{1,k+1} \vee x_{k+1,l} \vee \neg x_{1,l} \quad (6)$$

By applying Resolution between clause (3) and the transitivity clauses, we obtain

$$x_{1,2} \vee x_{1,3} \vee \dots \vee x_{1,k+1} \vee \neg x_{1,l}. \quad (7)$$

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 (7) 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 $O(k^2n) + 2^{O(k^2)}$, it is fpt-bounded. \square

5. DISCUSSION AND OPEN PROBLEMS

Is the parameterized proof system \widehat{P} from Definition 2.9 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 Dantchev et al. [2007] 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. Lower bounds for its parameterized version \widehat{F} seem even harder to achieve for strong contradictions (as we just add new axioms). A similar conclusion remains true if we combine all parameterized axioms into one (using, e.g., Buss [1987]) but allow arbitrary parameterized contradictions, not necessarily strong.

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 5.1. 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 Bonet et al. [1997] and Pudlák [1997]) and no direct, combinatorial proof is currently known. And if we try to generalize the methods from Bonet et al. [1997] and Pudlák [1997] (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 5.2. Find an explicit *partial* monotone function (a.k.a. a monotone promise problem) $f : [n]^{\leq k} \rightarrow \{0, 1\}$ defined 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, Dantchev et al. [2007, 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”

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

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

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

The disturbing Example 2.10 turns into an instance of PHP_k^{k+1} with large “metapipeons” that has an fpt-bounded proof (e.g., the straightforward adaption of the rectangular proof from Razborov et al. [2002, Example 1]). Thus, following Dantchev et al. [2007], we ask the following.

Question 5.3. 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 Beame et al. [2007], let G be a graph $[n]$ in which vertices are split into k subsets V_1, \dots, 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

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

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

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

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 Dantchev et al. [2007] (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 5.3.

Question 5.4. Do the principles $\alpha_{block}(G, k)$ always have fpt-bounded Resolution 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 Erdős-Rényi random graphs $G(n, p)$ for an appropriately chosen value of p . A lower bound for these formulas has been recently proved for tree-like Resolution by Beyersdorff et al. [2011a].

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 $S^{O(1)}$. Alekhovich and Razborov [2008] proved that if (classical) Resolution or tree-like Resolution were automatizable, then $W[P]$ would coincide with FPR, the randomized version of FPT. Eventually the derandomization in Eickmeyer et al. [2008] improved the result so that under the same automatizability assumptions $W[P]$ would coincide with FPT. On the other hand, tree-like Resolution is quasi-polynomially automatizable (see, e.g., Beame et al. [2002]).

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 $k \leq \log n$, this is smaller than $n^{\log n}$; otherwise, $\binom{n}{k+1} \leq 2^{(k+1)^2}$ 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 the following.

Question 5.5. 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 $(F, k) \in \text{PCon}$ 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

- ALEKHOVICH, M. AND RAZBOROV, A. A. 2008. Resolution is not automatizable unless $W[P]$ is tractable. *SIAM J. Comput.* 38, 4, 1347–1363.
- BEAME, P., KARP, R. M., PITASSI, T., AND SAKS, M. E. 2002. The efficiency of resolution and Davis–Putnam procedures. *SIAM J. Comput.* 31, 4, 1048–1075.
- BEAME, P., IMPAGLIAZZO, R., AND SABHARWAL, A. 2007. The resolution complexity of independent sets and vertex covers in random graphs. *Computat. Complex.* 16, 3, 245–297.
- BEYERSDORFF, O., GALESI, N., AND LAURIA, M. 2011a. Parameterized complexity of DPLL search procedures. In *Proceedings of the 14th International Conference on Theory and Applications of Satisfiability Testing*. Lecture Notes in Computer Science, vol. 6695, Springer-Verlag, Berlin, 5–18.
- BEYERSDORFF, O., GALESI, N., LAURIA, M., AND RAZBOROV, A. 2011b. Parameterized bounded-depth Frege is not optimal. In *Proceedings of the 38th International Colloquium on Automata, Languages, and Programming*. Lecture Notes in Computer Science, vol. 6755, Springer-Verlag, Berlin, 630–641.
- BONET, M. L. AND GALESI, N. 2001. Optimality of size-width tradeoffs for resolution. *Computat. Complex.* 10, 4, 261–276.
- BONET, M. L., PITASSI, T., AND RAZ, R. 1997. Lower bounds for cutting planes proofs with small coefficients. *J. Symb. Logic* 62, 3, 708–728.
- BUSS, S. R. 1987. Polynomial size proofs of the propositional pigeonhole principle. *J. Symb. Logic* 52, 916–927.
- BUSS, S. R. AND PITASSI, T. 1997. Resolution and the weak pigeonhole principle. *Comput. Sci. Logic*. 149–156.
- CHEN, Y. AND FLUM, J. 2008. The parameterized complexity of maximality and minimality problems. *Ann. Pure Appl. Logic* 151, 1, 22–61.
- COOK, S. A. AND RECKHOW, R. A. 1979. The relative efficiency of propositional proof systems. *J. Symb. Logic* 44, 1, 36–50.
- DANTCHEV, S. S., MARTIN, B., AND SZEIDER, S. 2007. Parameterized proof complexity. In *Proceedings of the 48th IEEE Symposium on the Foundations of Computer Science*. 150–160.
- DOWNNEY, R. G. AND FELLOWS, M. R. 1999. *Parameterized Complexity*. Springer-Verlag, Berlin.

- EICKMEYER, K., GROHE, M., AND GRÜBER, M. 2008. Approximation of natural W[P]-complete minimisation problems is hard. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity*. IEEE Computer Society Press, 8–18.
- FLUM, J. AND GROHE, M. 2003. Describing parameterized complexity classes. *Inf. Computat.* 187, 2, 291–319.
- FLUM, J. AND GROHE, M. 2006. *Parameterized Complexity Theory*. Springer-Verlag, Berlin Heidelberg.
- GAO, Y. 2009. Data reductions, fixed parameter tractability, and random weighted d-CNF satisfiability. *Artif. Intell.* 173, 14, 1343–1366.
- HAKEN, A. 1985. The intractability of resolution. *Theor. Comput. Sci.* 39, 297–308.
- KRAJÍČEK, J., PUDLÁK, P., AND WOODS, A. 1995. Exponential lower bounds to the size of bounded depth Frege proofs of the pigeonhole principle. *Random Struct. Algor.* 7, 1, 15–39.
- NIEDERMEIER, R. 2006. *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and Its Applications, Oxford University Press.
- PITASSI, T., BEAME, P., AND IMPAGLIAZZO, R. 1993. Exponential lower bounds for the pigeonhole principle. *Computat. Complex.* 3, 97–140.
- PUDLÁK, P. 1997. Lower bounds for resolution and cutting planes proofs and monotone computations. *J. Symb. Logic* 62, 3, 981–998.
- RAZ, R. 2004. Resolution lower bounds for the weak pigeonhole principle. *J. ACM* 51, 2, 115–138.
- RAZBOROV, A., WIGDERSON, A., AND YAO, A. 2002. Read-once branching programs, rectangular proofs of the pigeonhole principle and the transversal calculus. *Combinatorica* 22, 4, 555–574.
- RAZBOROV, A. A. 2004. Resolution lower bounds for perfect matching principles. *J. Comput. Syst. Sci.* 69, 1, 3–27.
- RIIS, S. 2001. A complexity gap for tree resolution. *Computat. Complex.* 10, 3, 179–209.

Received September 2011; revised March 2012; accepted July 2012