TFNP Intersections Through the Lens of Feasible Disjunction

Pavel Hubáček
Institute of Mathematics, Czech Academy of Sciences, Prague, Czech Republic
Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic

Erfan Khaniki
Institute of Mathematics, Czech Academy of Sciences, Prague, Czech Republic
Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic

Neil Thapen
Institute of Mathematics, Czech Academy of Sciences, Prague, Czech Republic

Abstract

The complexity class CLS was introduced by Daskalakis and Papadimitriou (SODA 2010) to capture the computational complexity of important TFNP problems solvable by local search over continuous domains and, thus, lying in both PLS and PPAD. It was later shown that, e.g., the problem of computing fixed points guaranteed by Banach's fixed point theorem is CLS-complete by Daskalakis et al. (STOC 2018). Recently, Fearnley et al. (J. ACM 2023) disproved the plausible conjecture of Daskalakis and Papadimitriou that CLS is a proper subclass of PLS $\cap$ PPAD by proving that CLS $=$ PLS $\cap$ PPAD.

To study the possibility of other collapses in TFNP, we connect classes formed as the intersection of existing subclasses of TFNP with the phenomenon of *feasible disjunction* in propositional proof complexity; where a proof system has the feasible disjunction property if, whenever a disjunction $F \vee G$ has a small proof, and F and G have no variables in common, then either F or G has a small proof. Based on some known and some new results about feasible disjunction, we separate the classes formed by intersecting the classical subclasses PLS, PPA, PPAD, PPADS, PPP and CLS. We also give the first examples of proof systems which have the feasible interpolation property, but not the feasible disjunction property.

2012 ACM Subject Classification Theory of computation → Proof complexity; Theory of computation → Problems, reductions and completeness; Theory of computation → Complexity classes

Keywords and phrases TFNP, feasible disjunction, proof complexity, TFNP intersection classes

Digital Object Identifier 10.4230/LIPIcs.ITCS.2024.63

Funding *Pavel Hubáček*: Supported by the Czech Academy of Sciences (RVO 67985840).
Erfan Khaniki: Supported by the Czech Academy of Sciences (RVO 67985840), GAČR grant 19-27871X and SVV-2023-260721.
Neil Thapen: Supported by the Czech Academy of Sciences (RVO 67985840) and GAČR grant 23-04825S.

Acknowledgements We would like to thank Lukáš Folwarczný, Tuomas Hakoniemi, Yuhao Li, William Pires, Pavel Pudlák and Robert Robere for helpful discussions related to this work.

1 Introduction

Since the foundational work of Megiddo and Papadimitriou [40], various subclasses of total search problems have been introduced to characterize the computational complexity of problems for which a solution is guaranteed to exist [32, 43, 36, 31, 49, 34, 30, 26, 35]. Such problems are naturally clustered based on the kind of reasoning, or "combinatorial lemma", used to prove that all instances have a solution; or viewed alternatively, based on the type of an inefficient algorithm that can be used to solve them. A prime example is the class PLS

© Pavel Hubáček, Erfan Khaniki, and Neil Thapen;
licensed under Creative Commons License CC-BY 4.0

of all total search problems solvable via a simple local search algorithm, which can also be defined as the class of problems for which reasoning by induction over NP formulas is enough to prove that a solution exists [9].

The complexity class CLS was introduced by Daskalakis and Papadimitriou [14] to capture the computational complexity of important TFNP problems solvable by local search over continuous domains and, thus, lying in both PLS and PPAD. It was later shown that, e.g., the problem of computing fixed points guaranteed by Banach's fixed point theorem is CLS-complete by Daskalakis et al. [15]. Recently, Fearnley et al. [18] disproved the plausible conjecture of Daskalakis and Papadimitriou that CLS is a proper subclass of $\mathsf{PLS} \cap \mathsf{PPAD}$ by establishing that $\mathsf{CLS} = \mathsf{PLS} \cap \mathsf{PPAD}$. Their work motivates the natural question whether there will be other surprising collapses of TFNP intersections; as noted by Fearnley et al. [18], CLS is currently the only intersection class that has natural complete problems in the sense that input is explicit and not represented succinctly via a Boolean circuit.

Among the well-studied subclasses of TFNP, a prime candidate for being an intersection class is PPAD which is contained in the intersection of all the PPA_q classes as well as PPADS. As for other intersections of classical subclasses of TFNP, Göös et al. [21] gave simple combinatorial complete problems for the intersections of PLS with PPADS and PPP (respectively). Ishizuka [27] gave a combinatorial characterization of $\mathsf{PLS} \cap \mathsf{PPA}$.

TFNP and proof complexity. Our understanding of the structural properties of subclasses of TFNP is largely based on its interaction with proof complexity. The seminal paper of Beame et al. [2] demonstrated that lower bounds in proof complexity can illuminate the relative complexity of TFNP subclasses, which was subsequently developed further [41, 42, 4]. Recent work pushed this connection further and characterized various subclasses of TFNP in terms of corresponding proof systems, thus leading to oracle separations between various subclasses of TFNP when conflicting bounds were known for the corresponding proof systems [22]. In this work, we build upon the recent characterizations of subclasses of TFNP. We take a generic approach to intersections of standard subclasses of TFNP employing the classical notion of *feasible disjunction* in proof complexity.

Suppose F and G are propositional formulas with no variables in common. Then if the disjunction $F \vee G$ is a tautology, it follows that either F is a tautology or G is. Thus a natural property that a proof system may have is that, given a proof π of $F \vee G$ in the system, there exists another proof, of a similar length to π or shorter, of either F or G by itself. This property was called *weak feasible disjunction* by Pudlák [46] (see also [38, 48]); in this paper we call it simply *feasible disjunction*[1]. It has been shown to hold for proof systems such as resolution and cutting planes [44], Lovász-Schrijver with the rounding rule [45], and polynomial calculus and sum-of-squares [24]. Up until now, the only negative result is recent work by Garlík [19] showing that for $k \geq 2$ the system $\mathrm{Res}(k)$, a small extension of resolution, does not have feasible disjunction. Beyond [19], nothing was known (in classical logic), even conditional on some hardness assumption. In particular, it is widely open for strong proof systems such as Frege and extended Frege (more is known for non-classical logic, see e.g. [10], [29]).

A related property of a proof system is *feasible interpolation* [37]. This holds if, given a proof of $F \vee G$, where F and G may share variables, one can construct a small circuit which, from any assignment to the common variables, computes which of F and G is a tautology

[1] "Feasible disjunction", without the word "weak", is also used in the literature to mean the stronger property, that there is a polynomial time algorithm which produces a proof of either F or G from a proof of $F \vee G$.

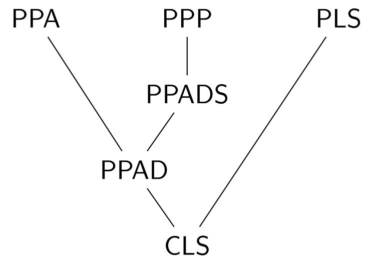

Figure 1 The relative complexity of the standard subclasses of TFNP.

under this assignment. This has been intensively studied because one can often guarantee that the circuit is monotone, and thus obtain proof complexity lower bounds from monotone circuit lower bounds. It was observed by Pudlák [46] that systems known to have feasible disjunction also have feasible interpolation, and vice versa; this remains true for the more recent work mentioned above, largely because the proof method for one typically also gives the other.

The proof systems we use below are more naturally used to refute contradictions, rather than to prove tautologies. For this reason when we use feasible disjunction, we actually use the "dual" version of it which is appropriate for such systems, involving conjunction rather than disjunction: if $F \wedge G$ has a small refutation, then either F has a small refutation, or G does.

1.1 Our Results and Technical Overview

Below, we give a high-level discussion of our results and techniques. Detailed definitions and the necessary background are provided in Section 2 and the formal details of our results in the subsequent Sections 3–5.

1.1.1 Separation of TFNP intersections (Section 4)

Consider the six best-studied TFNP classes PLS, PPA, PPP, PPADS, PPAD, and CLS depicted in Figure 1. We show that, other than the equality $\mathsf{CLS} = \mathsf{PLS} \cap \mathsf{PPAD}$, no nontrivial intersection of two of these classes equals a third, and no two nontrivial intersections are equal to each other (Theorem 32). This remains true if we include also all classes PPA_q, for all primes q (Theorem 33). Our method relies on the well-known connection between natural TFNP classes and proof systems (in particular, as it is set out in [22]).

For a suitable family Q_n of contradictory CNF formulas, we write $\mathsf{Search}(Q_n)$ for the search problem of finding a false clause in Q_n, given an assignment. For a natural TFNP class A, there typically exists a proof system P such that $\mathsf{Search}(Q_n) \in A$ if and only if Q_n has small refutations in P, for some appropriate meaning of "small". Now suppose that $B, C \not\subseteq A$ are TFNP classes and $B \cap C \subseteq A$. Take CNF families Q and R, in disjoint variables, such that $\mathsf{Search}(Q_n) \in A$ and $\mathsf{Search}(R_n) \in B$. It is easy to see that $\mathsf{Search}(Q_n \wedge R_n) \in B \cap C$, and, thus, $Q_n \wedge R_n$ has small refutations in the system P. Therefore, if P has a suitable form of feasible disjunction, we know that for each n either Q_n has a small refutation in P, or R_n does. Now if we had the stronger conclusion that either (Q_n has small refutations for every n) or (R_n does, for every n), and we had chosen Q and R to be "complete" for the classes B and C in some appropriate sense, it would follow that either $B \subseteq A$ or $C \subseteq A$, contradicting our assumption on B and C. In fact, whether it is Q_n or R_n can in principle be different for each n, so we use more ad hoc arguments which directly use proof complexity lower bounds for Q_n and/or R_n in P to get a contradiction (where these may be the same lower bounds that were used to show $B, C \nsubseteq A$ in the first place).

We note that [16] used a similar approach, implicitly relying on the feasible disjunction property, to show (independently of our work) that a certain TFNP intersection is not in PLS. The independent work of Li, Pires, and Robere [39] develops some new characterizations of intersection classes by proof systems.

1.1.2 Feasible disjunction (Section 3)

We define the feasible disjunction property for a general measure μ of proof complexity:

▶ **Definition 1.** *Let P be a proof system and let μ be a measure for P. We say that P has* μ-feasible disjunction, *or* feasible disjunction for μ, *if, for CNFs F, G in disjoint variables,*

$$\min(\mu(F), \mu(G)) \leq \mathsf{poly}(\mu(F \wedge G), \log n)$$

where n is the total number of variables in F and G.

For our results on intersections, we use that the Nullstellensatz proof system has feasible disjunction for degree, over any field, and that the Sherali-Adams system has feasible disjunction for the measure μ^*_{SA}, defined as log(magnitude) + degree. The first result appears already in [22] and the second is new (Corollary 26).

Both things are proved in a similar way: if a CNF F has no small proofs, then there exists an object that behaves like an approximation of an assignment that satisfies F. If G also has no small proofs, we can combine the object for F with the object for G to get something that behaves like an approximation of an assignment satisfying $F \wedge G$, exploiting the fact that F and G share no variables. This is then enough to show that $F \wedge G$ has no small proofs. Hakoniemi [25] used this approach to show feasible disjunction for degree for Sherali-Adams, and feasible disjunction for combined degree and magnitude for the sum-of-squares proof system. The same approach can show feasible disjunction for width in the resolution proof system, as in [16].

For our result for Sherali-Adams, we develop a new "semantic" characterization of Sherali-Adams in small magnitude and degree, modelled on the result for sum-of-squares in [25]. Precisely, we define an *ε-pseudoexpectation* over degree d for a set of polynomials Q and show that such an object exists, for a small error ε, if and only if Q has no degree-d Sherali-Adams refutation of small magnitude (Lemmas 21 and 23). We note that a characterization of *size* in unary Sherali-Adams of a similar nature will appear in [17].

Finally we return to the standard meaning of feasible disjunction, which is for the size measure. We show that neither of the systems *reversible resolution* and *reversible resolution with terminals* has feasible disjunction in this sense (Lemma 30). This is interesting because both of them have feasible interpolation, which they inherit as subsystems of resolution [37]. As far as we know, these are the first examples of systems with feasible interpolation but not feasible disjunction, addressing the issue raised by Pudlák [46]. Our proof uses known upper and lower bounds from proof complexity, together with the characterization from [22] of RevResT as a kind of intersection of resolution and unary Nullstellensatz.

1.1.3 Combinatorial characterization of PPA ∩ PPADS (Section 5)

Our separations, in particular, show that $\mathsf{PPAD} \subset \mathsf{PPA} \cap \mathsf{PPADS} \subset \mathsf{PPA} \cap \mathsf{PPP}$, i.e., the two intersections $\mathsf{PPA} \cap \mathsf{PPADS}$ and $\mathsf{PPA} \cap \mathsf{PPP}$ that were not explicitly studied in the literature yet might be of independent interest. As a first step towards improving our understanding of these intersection classes, we introduce a complete combinatorial problem for PPA ∩ PPADS. The new total search problem, which we call Leaf-Or-Sink, is close to End-Of-Line and other similar combinatorial problems characterizing the standard classes PPA, PPAD, and PPADS. It is an interesting open problem to provide a similar combinatorial problem for the higher intersection PPA ∩ PPP. In particular, it is known that (under randomized reductions) PPA ∩ PPP contains integer factoring [5, 30], and, thus, a concise complete problem for PPA ∩ PPP could, for example, help in characterizing the search complexity of integer factoring.

For the formal definition of Leaf-Or-Sink, see Definition 35. Here, we give an informal description of the problem. Recall that an instance of End-Of-Line is given via a succinctly represented directed graph with vertices of in-degree and out-degree at most one (i.e., a collection of directed paths and cycles) with a known distinguished source. A solution is any sink in the graph or a source distinct from the distinguished source. Our PPA ∩ PPADS-complete problem Leaf-Or-Sink is defined on a generalization of an End-Of-Line graph. The instance is given via a succinctly represented directed graph, where each vertex has at most two neighbors (i.e., we allow vertices of in-degree or out-degree two) with a distinguished source of degree one. A solution is any sink (of degree one or two) or a leaf (i.e., a vertex with a single neighbor) distinct from the distinguished source.

The containment of Leaf-Or-Sink in PPA∩PPADS follows via straightforward reductions to the PPA-complete problem Leaf (i.e., the undirected variant of End-Of-Line) and PPADS-complete problem Sink (i.e., the variant of End-Of-Line where sources do not constitute solutions). We establish the hardness for PPA ∩ PPADS via a reduction from the canonical PPA ∩ PPADS-complete problem, where we are given an instance of Leaf and an instance of Sink and we are asked to solve either one of them. Our reduction orients the edges of the Leaf instance arbitrarily and resolves the "useless" solutions at sinks of degree two that are unrelated to leaves of the Leaf instance (that are likely introduced by the arbitrary orientation of the edges) using the instance of Sink. On a high level, for each predecessor of any sink of degree two, we can create a copy of the Sink graph and, instead of the sink of degree two, connect the predecessor to the distinguished source in its copy of the Sink graph. In this way, any sink of in-degree two becomes isolated and is no longer a solution to Leaf-Or-Sink. Unfortunately, even though we have resolved sinks of in-degree two, the copies of the Sink graph might have added new sources of degree one, which would constitute a solution to Leaf-Or-Sink but not to Sink. However, it is easy to take care of these sources due to the fact that there are exactly two copies of the Sink graph for each sink of in-degree two. For each sink of degree two that we were resolving, we can simply add a new source of degree two with outgoing edges to the two copies of the source. By such "gluing" of the Leaf graph with the copies of the Sink graph, we would construct an instance of Leaf-Or-Sink without sinks of degree two and where the leaves are either leaves in the Leaf graph or sinks in the Sink graph.

2 Preliminaries

2.1 Proof Systems and Measures

We recall some standard definitions. A *clause* is a disjunction of propositional literals, where we write $\bar{x}_i$ for the negation of the variable x_i. A CNF is a conjunction of clauses. A refutation of a CNF F in a proof system P is a witness that F is unsatisfiable, where the form of witness that is allowed depends on the system P.

▶ **Definition 2.** *A* measure μ *for a proof system* P *is a function which assigns a non-negative real number* $\mu(\pi)$ *to each* P*-refutation* π*.*

For a CNF F *we define* $\mu(F) := \min\{\mu(\pi) : \pi \text{ is a } P\text{-refutation of } F\}$*.*

The standard measure for a system is the size (for example in bits) of a refutation, and "feasible" proofs typically mean those of polynomial size in the size of the CNF F. However in this paper, following e.g. [22], we use a different scaling, using for example the logarithm of size, rather than size, and typically working with polylogarithmic measures (an exception is Section 3.3).

We work mostly with the algebraic systems Nullstellensatz and Sherali-Adams and their variants, which we introduce now. We also use two variants of the standard propositional proof system resolution, which we describe at the end of this section.

In the algebraic systems we work with polynomials in some field $\mathcal{F}$ over some set of variables $x_1, \dots, x_n$, together with *twin/negated variables* $\bar{x}_1, \dots, \bar{x}_n$. Usually we constrain each x_i to be 0/1-valued and constrain $\bar{x}_i$ to take the opposite value. With this in mind we define I_n to be the ideal generated by the *Boolean axioms* $x^2 - x$ and the *negation axioms* $1 - x - \bar{x}$. We write $p \equiv q$ to mean that polynomials p and q are equal modulo this ideal.

A *monomial* is a product of variables, without any coefficient. We write S_d for the set of all monomials of degree d or less and $\mathcal{F}[S_d]$ for the set of polynomials of degree d or less, which we sometimes think of as a vector space spanned by S_d. We sometimes use the bound $|S_d| \leq (3n)^d$. When working over the reals, we write $\mathbb{R}^+[S_d]$ for the set of polynomials in $\mathbb{R}[S_d]$ which only have positive coefficients. For a real polynomial p, we write $||p||$ for the maximum absolute value of any coefficient in p.

Over the ideal I_n we can convert any CNF F into a set of polynomials Q expressing the same thing as F, by translating a clause C into the polynomial equation $\prod_{j \in J} \bar{x}_j \prod_{k \in K} x_k = 0$ where J is the set of variables appearing positively in C and K is the set of variables appearing negatively. In this way we can use algebraic proof systems to refute CNFs.

▶ **Definition 3.** *Let* $\mathcal{F}$ *be a field and let* Q *be a set of polynomials over* $\mathcal{F}$*. A* nullstellensatz refutation *over* $\mathcal{F}$*, or* $\mathsf{NS}_\mathcal{F}$ refutation*, of* Q *is an identity*

$$1 \equiv \sum_{q \in Q} t_q q$$

where each t_q *is a polynomial over* $\mathcal{F}$*.*

Our main measure for an $\mathsf{NS}_\mathcal{F}$ refutation is *degree*, sometimes written as deg, which we define to be the maximum degree of the expressions $t_q q$. Note that we do not explicitly include the degree of the element of I_n hidden in the notation $\equiv$ in this measure; but we may assume without loss of generality that it is bounded by the degree of the expressions $t_q q$, by the following argument. Suppose $p \equiv p'$ where p' is the multilinearization of p. Suppose we multilinearize p term-by-term, starting with some term rx^2 which we replace with rx. This is the same as adding $-r(x^2 - x)$ from the ideal, and to do this we never need an element of the ideal with degree higher than $\mathsf{deg}(p)$. A similar argument works for dealing with the negation axioms.

Another natural measure we could consider here is the log of the monomial size of a refutation, where monomial size counts the total number of monomials that appear. However, since we can bound the number of monomials in a polynomial by $|S_d| \leq (3n)^d$, from our point of view this would be essentially the same as the degree measure.

▶ **Definition 4.** *Let Q be a set of real polynomials. A* Sherali-Adams refutation*, or* SA refutation*, of Q is an identity*

$$-1 \equiv v + \sum_{q \in Q} t_q q$$

where v and the t_q are real polynomials, and v only has positive coefficients.

A unary Sherali-Adams *or* uSA *refutation is an* SA *refutation in which only integer coefficients are allowed.*

The *degree* of such a refutation is the maximum of the degrees of the expressions $t_q q$ and v. The *magnitude* is the maximum of the absolute values of the coefficients appearing in these expressions. The *size* of an SA refutation is the total number of monomials that appear (counting repetitions). The *unary size* of a uSA refutation is the sum of the magnitudes of the coefficients in front of all the monomials. We also define combined measures for SA as

$$\mu^*_{\mathsf{SA}}(\pi) := \log(\text{magnitude of } \pi) + \mathsf{deg}\ \pi$$
$$\mu_{\mathsf{uSA}}(\pi) := \log(\text{unary size of } \pi) + \mathsf{deg}\ \pi$$

where $\mu_{\mathsf{uSA}}(\pi)$ is only defined if π is a uSA refutation. Note that for our purposes, the sum in each measure could be replaced with the maximum, with essentially no change.

Turning to purely propositional systems, a *resolution* refutation of a CNF F is a sequence of clauses, concluding with the empty clause, in which each clause is either from F, or follows from earlier clauses either by the resolution rule, which derives $C \vee D$ from $C \vee x$ and $D \vee \bar{x}$, for any variable x, or by the weakening rule, which derives any clause $D \supseteq C$ from C. The *width* of such a refutation is the size of the largest clause, and is analogous to the degree measure for algebraic systems. In Section 3.3 we need the following variants of resolution, introduced in [22]. For these, we only use the standard size measure.

A *reversible resolution*, or RevRes, refutation of a CNF F is a sequence of multisets $M_1, \ldots, M_t$ of clauses, where M_1 contains only clauses in F; M_t contains the empty clause; and each M_{i+1} is derived from M_i either by the *reversible resolution rule*, which deletes one clause of the form $C \vee x$ and one clause of the form $C \vee \bar{x}$ from M_i, replacing them both with a single clause C; or by the *reversible weakening rule*, which is the same thing in reverse.

A *reversible resolution with terminals*, or RevResT, refutation of F is a RevRes refutation of F with the extra condition that M_t contains exactly one copy of the empty clause and every other clause in M_t is a weakening of a clause of F (in the usual sense of weakening).

2.2 Decision-Tree TFNP and Proof Complexity

We work with the non-uniform "decision tree" model of TFNP and model our definitions and notation on [22]. A slight exception is Section 5, where we define a new class in terms of graphs given by circuits, without any explicit reference to an oracle; but this can be made into a definition in the decision-tree style without any difficulties.

▶ **Definition 5.** *A* total search problem *is a sequence $R = (R_n)_{n \in \mathbb{N}}$ of relations $R_n \subseteq \{0,1\}^{m_n} \times \mathcal{O}_n$, where $m_n \in \mathbb{N}$, each $\mathcal{O}_n$ is a finite set, and for every* input *$x \in \{0,1\}^{m_n}$ there is some* solution *$y \in \mathcal{O}_n$ such that $(x, y) \in R_n$.*

▶ **Definition 6** (TFNP^{dt})**.** *A total search problem, defined as above, is in TFNP^{dt} if m_n is at most quasipolynomial in n and, for each $y \in \mathcal{O}_n$, there is a decision tree T_y of depth* $\mathsf{poly}(\log n)$*, querying x and deciding whether $(x, y) \in R_n$.*

▶ **Definition 7.** *Let $R_m \subseteq \{0,1\}^r \times \mathcal{O}_R$ and $S_n \subseteq \{0,1\}^s \times \mathcal{O}_S$ be two total search problems. A* reduction *of R_m to S_n is a pair of functions $f : \{0,1\}^r \to \{0,1\}^s$ and $g : \{0,1\}^r \times \mathcal{O}_S \to \mathcal{O}_R$ satisfying, for all $x \in \{0,1\}^r$ and all $y' \in \mathcal{O}_S$,*

$$(f(x), y') \in S_n \longrightarrow (x, g(x, y')) \in R_m.$$

The reduction has depth d *if each bit of $f(x)$ and for every $y' \in \mathcal{O}_S$, the function $x \mapsto g(x, y')$ is computable by a depth-d decision tree.*

▶ **Definition 8.** *Let R and S be total search problems in TFNP^{dt}. We say R* is reducible to *S if for each n there is m quasipolynomial in n such that there is a depth-$\mathsf{poly}(\log n)$ reduction of R_n to S_m. We write this as $R \leq S$. If also $S \leq R$ we say R and S are* equivalent.

This expresses in our non-uniform setting the idea that we can use solutions to S to find solutions to R. Note that $R \leq S$ is equivalent to the condition $S^{dt}(R) = \mathsf{poly}(\log n)$ of [22]. Unless stated otherwise, we use only the model of TFNP described above, so we omit the superscripts ^{dt} in the names of classes that would otherwise be used to indicate this.

▶ **Definition 9.** *A* narrow family of CNFs *is a sequence $F = (F_n)_{n \in \mathbb{N}}$ in which in each F_n the number of variables and clauses is quasipolynomial in n, and the width is polylogarithmic in n.*

As usual, we sometimes use the notation like "a narrow CNF F_n" to mean a family.

▶ **Definition 10.** *Given an unsatisfiable CNF G, define* $\mathsf{Search}(G)$ *to be the problem of finding a false clause in G, given a total assignment to the variables of G.*

Now let $F = (F_n)_{n \in \mathbb{N}}$ be a narrow family of unsatisfiable CNFs. Then the sequence $(\mathsf{Search}(F_n))_{n \in \mathbb{N}}$ is in TFNP.

▶ **Definition 11.** *Let P be a proof system and let μ_P be a measure for P. We say that a* TFNP *class A* is characterized by *P* under *μ_P if, for every narrow family $(F_n)_{n \in \mathbb{N}}$ of unsatisfiable CNFs, we have $(\mathsf{Search}(F_n))_{n \in \mathbb{N}} \in A$ if and only if $\mu_P(F_n) \leq \mathsf{poly}(\log n)$.*

We use the word "characterized" because it is possible to present any TFNP problem as a false clause search problem of this kind, as follows.

▶ **Lemma 12.** *Let $(R_n)_{n \in \mathbb{N}}$ be a* TFNP *problem. Let $\mathsf{CNF}(R_n)$ be a CNF asserting that x is an input with no solution in R_n – that is, it asserts that for every y in the solution space, the decision tree for y does not accept x. Then $\mathsf{CNF}(R_n)$ is a narrow family of unsatisfiable CNFs and, as* TFNP *problems, $\mathsf{Search}(\mathsf{CNF}(R_n))$ and R_n are equivalent.*

We list some connections between TFNP classes and proof systems; see [22] for a more complete list.

▶ **Proposition 13.**

1. PPA_q *is characterized by $\mathsf{NS}_{\mathbb{F}_q}$ under degree, for each prime q [20, 33]*
2. PPADS *is characterized by* uSA *under μ_{uSA} [22]*
3. PLS *is characterized by resolution under width [8].*

2.3 Some Upper and Lower Bounds

We describe three families of unsatisfiable, narrow CNFs we use.

The *bit pigeonhole principle* BPHP_n has variables for the bits of $2^k + 1$ binary strings, each of length k, where $k = \lfloor \log n \rfloor$. It consists of a clause for each pair of strings, asserting that those two strings are distinct. We think of total assignments to these variables as maps from $[n+1]$ pigeons (the indices of the strings) to $[n]$ holes (the values of the strings), so the principle asserts that no two pigeons go to the same hole.

▶ **Proposition 14.**
1. *The problem* $\mathsf{Search}(\mathrm{BPHP}_n)$ *is in* PPP
2. BPHP_n *requires degree* n *to refute in* SA.

Proof. For 1, we use that we have essentially a 1-1 correspondence between the solutions of the usual pigeonhole principle search problem and the clauses of BPHP_n.

Item 2 is shown (implicitly) in [12]. Their way of encoding CNFs into SA refutations is slightly different from ours, since they translate clauses into linear inequalities rather than monomial equations, so we confirm that their proof works with our encoding. We translate the principle BPHP_n into a system of equations $m_{i,i',j} = 0$, for each pair i, i' of distinct pigeons and each hole j, where $m_{i,i',j}$ is the natural monomial, of degree $2k$, which is 1 if and only if both i and i' go to hole j. Suppose we have a SA refutation of this system of degree at most n. We may write it as an equality

$$-1 = v + \sum_{i,i',j \;\, i \neq i'} t_{i,i',j} m_{i,i',j} + B$$

where v is a sum of monomials with positive coefficients, B is in the ideal generated by the Boolean and negation axioms, and everything has degree at most n. We define an evaluation $E : S_n \to \mathbb{R}$ as follows. For any $m \in S_n$, let X be the set of pigeons whose variables appear in m. Extend X arbitrarily to a set X' of exactly n pigeons, and let $E(m)$ be the probability that m evaluates to 1 under a random matching of X' to the set $[n]$ of all holes. Observe that $E(m)$ is independent of the choice of X'. We extend E to a linear map $\mathbb{R}[S_n] \to \mathbb{R}$ by linearity.

Then $E(1) = 1$, since the empty monomial is always satisfied. Also $E(v) \geq 0$, and for each i, i', j in the sum we have $E(t_{i,i',j} m_{i,i',j}) = 0$, since every matching with i and i' in its domain sets $m_{i,i',j}$ to 0. Finally let z be any single variable and let m be any monomial of suitable degree. Then $E(mz^2) = E(mz)$ by construction, and also $E(mz) + E(m\bar{z}) = E(m)$, since we may assume that the pigeon associated with z is in X', so every matching under consideration which satisfies m satisfies exactly one of mz and $m\bar{z}$. Thus $E(B) = 0$. Together these contradict the displayed equality above. (Such an E is called in the literature a *pseudoexpectation*.) ◀

The *sink-of-DAG principle* SoD_n has variables for a directed graph on $[n]$, with a distinguished source node, such that edges only go from smaller to larger nodes in $[n]$. It asserts that this graph must have a sink.

▶ **Proposition 15.**
1. *The problem* $\mathsf{Search}(\mathrm{SoD}_n)$ *is in* PLS
2. *For any prime* q*, any* $\mathsf{NS}_{\mathbb{F}_q}$ *refutation of* SoD_n *requires degree* $\Omega(n)$.
3. *Any* $n^{o(1)}$ *degree* SA *refutation of* SoD_n *has exponential magnitude.*

Proof. Item 1 is straightforward. Item 2 follows from lower bounds in [11, 6] on the closely-related house-sitting principle. Item 3 is one of the main results of [22], but we must show that it still holds with our definitions, which are not quite the same – in particular the version of SA in [22] does not have negated variables, and they count the coefficients in the ideal I_n towards the magnitude.

Let Q be the set of polynomials encoding SoD_n. Suppose we have an SA refutation of Q of degree $d \in n^{o(1)}$ and magnitude R, of the form

$$-1 \equiv v + \sum_{q \in Q} t_q q.$$

We let a and b be the polynomials respectively in the ideal of Boolean axioms and the ideal of negation axioms witnessing this equivalence. So we have the following equation, where we may assume that any polynomial appearing in it has degree at most d.

$$\sum_{q \in Q} t_q q + a + b = 1 + v.$$

Now we get rid of the negated variables by replacing each occurrence of $\bar{x}_i$ with $(1 - x_i)$. After this transformation b becomes 0, and we can write the result as

$$\sum_{q \in Q} t'_q q' + a' = 1 + v'. \tag{1}$$

By the proof of Lemma 3 of [22] there is a total assignment to the variables of (1) under which the right-hand-side $1 + v'$, and thus also $1 + v$, evaluates to at least 1.4^n. But the maximum possible value of v is $R(3n)^d$, since v contains at most $(3n)^d$ monomials. Therefore R is exponential. ◀

For a prime p, the *mod-p counting principle* Count_n^p has variables describing a partition of $[pn + 1]$. It asserts that every set in this partition has size precisely p.

▶ **Proposition 16.**
1. *The problem* $\mathsf{Search}(\mathrm{Count}_n^q)$ *is in* PPA_q*, or in* PPA *for* $q = 2$.
2. *For distinct primes* p, q*, any* $\mathsf{NS}_{\mathbb{F}_q}$ *refutation of* Count_n^p *requires degree* $n^{\Omega(1)}$.

Proof. Item 1 is straightforward. Item 2 is from [7]. ◀

3 Feasible disjunction

We show in this section that Nullstellsatz has feasible disjunction for degree; that Sherali-Adams has feasible disjunction for the measure μ^*_{SA}; and that feasible disjunction for size fails for RevRes and $\mathsf{RevResT}$. What we would really like to show in the second case is that uSA has feasible disjunction for the measure μ_{uSA}, but the result we prove is enough for our argument about TFNP intersections, since the relevant lower bounds are on magnitude rather than explicitly on unary size.

3.1 Feasible Disjunction for Degree for Nullstellensatz

This is shown in Lemma 2, Claim 2 of [22], but we include a proof for completeness. We first need a “semantic” characterization of systems of polynomials with no low degree refutation.

▶ **Proposition 17** ([6]). *Let $\mathcal{F}$ be a field. Then a set of polynomials Q does not have $\mathsf{NS}_{\mathcal{F}}$-refutations of degree d if and only if there is a d-design for Q, that is, a linear function $D : \mathcal{F}[S_d] \to \mathcal{F}$ satisfying firstly that $D(1) = 1$ and secondly that $D(qr) = 0$ for q, r polynomials with $\mathsf{deg}(q) + \mathsf{deg}(r) \leq d$, where q is either an axiom from Q or a Boolean or negation axiom, and r is any polynomial.*

▶ **Theorem 18** ([22]). *For any field $\mathcal{F}$, $\mathsf{NS}_{\mathcal{F}}$ has deg-feasible disjunction.*

Proof. Suppose $Q(\vec{x})$ and $P(\vec{y})$ are two families of polynomials over disjoint variables $\vec{x}$ and $\vec{y}$ and moreover that $\mathsf{NS}_{\mathcal{F}}$ has a degree d refutation of $Q \cup P$. For the sake of contradiction assume neither Q nor P has an $\mathsf{NS}_{\mathcal{F}}$ refutation of degree d. By Proposition 17 degree d-designs D_Q and D_P exist for Q and P respectively.

Any $r \in \mathcal{F}[\vec{x}, \vec{y}]$ can be written uniquely as a polynomial $\sum_{i=1}^{k} a_i m_0^i m_1^i$ where each $m_0^i \in \mathcal{F}[\vec{x}]$ and $m_1^i \in \mathcal{F}[\vec{y}]$. We define a map from such polynomials, of degree at most d, to $\mathcal{F}$ as follows:

$$D(r) := \sum_{i=1}^{k} a_i D_Q(m_0^i) \cdot D_P(m_1^i).$$

We claim that D is a d-design for $Q \cup P$. It is clear from the definition that D is linear and $D(1) = 1$. For the second property, let q be an axiom and let $r \in \mathcal{F}[\vec{x}, \vec{y}]$ be as above, such that $\mathsf{deg}(q) + \mathsf{deg}(r) \leq d$. By the assumptions, q contains either only x variables or only y variables. Let us assume it is x variables; the case for y variables is similar. From the definition of D we obtain

$$D(qr) = \sum_{i=1}^{k} a_i D_Q(qm_0^i) \cdot D_P(m_1^i).$$

Since D_Q is a d-design for Q, we get that $D_Q(qm_0^i) = 0$ for each i, which implies that $D(qr) = 0$, as required. Thus D is a d-design for $Q \cup P$, which implies that $Q \cup P$ does not have $\mathsf{NS}_{\mathcal{F}}$ refutations of degree d, contradicting our initial assumption. ◀

3.2 Feasible Disjunction for Size and Degree for Sherali-Adams

We show μ^*_{SA}-feasible disjunction for SA. We base our approach closely on the proof of feasible disjunction for sum-of-squares in [25]. We need the notion of an SA derivation:

▶ **Definition 19.** *An* SA derivation *of a polynomial p from a set of polynomials Q is a polynomial identity*

$$p \equiv v + \sum_{q \in Q} t_q q$$

where v only has positive coefficients. We think of such an expression as a witness that $p \geq 0$ on every solution $\vec{x}$ of Q. A refutation *of Q is then a derivation of -1 from Q.*

Again the degree *of the derivation is the maximum degree of the expressions on the right hand side, without any cancellations. We say that the derivation is R-*bounded *if $||v|| \leq R$ and $||t_q|| \leq R$ for every $q \in Q$, in other words, if it has magnitude bounded by R.*

Note that we do not bound the coefficients appearing in the polynomial in the ideal I_n which witnesses the equivalence $\equiv$ in the definition of R-boundedness.

▶ **Definition 20.** *For $\varepsilon > 0$, an* ε-pseudoexpectation for Q over degree d *is a linear function* $E : \mathbb{R}[S_d] \to \mathbb{R}$ *satisfying*

1. $E(1) = 1$
2. $E(p) = E(q)$ *if* $p \equiv q$
3. $E(m) \geq -\varepsilon$ *for any* $m \in S_d$
4. $|E(mq)| \leq \varepsilon$ *for any* $m \in S_d$ *and* $q \in Q$.

We show now that the existence of an ε-pseudoexpectation over degree d is a semantic characterization of when Q has no degree-d SA refutations of small magnitude, and thus can play a similar role to how d-designs characterized degree-d $\mathsf{NS}_{\mathcal{F}}$ in Section 3.1.

▶ **Lemma 21** (Soundness). *Over degree d, if there is an ε-pseudoexpectation for Q, then there is no R-bounded refutation of Q for $R < 1/2\varepsilon|Q|(3n)^d$.*

Proof. Let E be an ε-pseudoexpectation for Q over degree d and suppose

$$-1 \equiv v + \sum_{q \in Q} t_q q$$

is a degree-d SA refutation of Q with coefficients bounded by $R < 1/2\varepsilon|Q|(3n)^d$. By items 1 and 2 in Definition 20, we have $E(v) + E(\sum t_q q) = -1$.

Recall v is a sum $\sum \lambda_i m_i$, where each $m_i \in S_d$ and $0 < \lambda_i < R$. Since $E(m_i) \geq -\varepsilon$ we have $E(\lambda_i m_i) \geq -R\varepsilon$, and thus $E(v) \geq -(3n)^d R\varepsilon > -1/2|Q|$, since $(3n)^d$ is a bound on the size of S_d. Similarly each t_q is the sum of at most $(3n)^d$ monomials, so $|E(t_q q)| \leq R(3n)^d \varepsilon < 1/2|Q|$, and thus in particular $E(\sum_{q \in Q} t_q q) > -1/2$. It follows that $E(v) + E(\sum t_q q) > -1$, a contradiction. ◀

▶ **Lemma 22.** *For any $m \in S_d$ there is a sum v of at most d monomials in S_d, in which every coefficient is* 1*, such that $1 - m \equiv v$.*

Proof. We prove this by induction. The base case is simply $1 - x_i \equiv \bar{x}_i$. For the inductive step, suppose $1 - m \equiv v$. Then

$$1 - mx_i \equiv \bar{x}_i + x_i - mx_i = \bar{x}_i + x_i(1 - m) \equiv \bar{x}_i + x_i v.$$

We may assume that m is multilinear, so that we need to consider each variable at most once. ◀

▶ **Lemma 23** (Completeness). *If there is no R-bounded refutation of Q of degree d, then there is a $(1/R)$-pseudoexpectation for Q over degree d.*

Proof. Suppose there is no R-bounded refutation of Q of degree d. Define sets $A, B \subseteq \mathbb{R}[S_d]$ by

$$A := \{p : p \equiv v \text{ some } v \in \mathbb{R}^+[S_d] \text{ with } ||v|| \leq R\}$$
$$B := \{-1 + \textstyle\sum_{q \in Q} t_q q : \ t_q q \in \mathbb{R}[S_d] \text{ with } ||t_q|| \leq R \text{ for all } q\}.$$

Then A and B are disjoint, since otherwise there would be an R-bounded refutation of Q of degree d. Furthermore they are both nonempty and, considered as subsets of the vector space $\mathbb{R}[S_d]$, are both convex.

Now let K be the quotient map $\mathbb{R}[S_d] \to \mathbb{R}[S_d]/I_n$. Then $K[A]$ and $K[B]$ are still nonempty and convex and, by the $\equiv$ in the definition of A, even disjoint. Thus by the hyperplane separation theorem in $\mathbb{R}[S_d]/I_n$ there is a nontrivial linear function $L' : \mathbb{R}[S_d]/I_n \to \mathbb{R}$

and a scalar $e \in \mathbb{R}$ such that $L'(p) \geq e$ for $p \in K[A]$ and $L'(p) \leq e$ for $p \in K[B]$. Composing K with L', we obtain a linear function $L : \mathbb{R}[S_d] \to \mathbb{R}$ with $L(p) \geq e$ for $p \in A$ and $L(p) \leq e$ for $p \in B$, with the extra property that $L(p) = L(q)$ if $p \equiv q$.

Since $0 \in A$ we have $e \leq L(0) = 0$ and since $-1 \in B$ we have $L(-1) \leq e$ and thus $L(1) \geq -e$. We claim $L(1) > 0$. If $e < 0$ this is clear, so suppose $L(1) = 0$ and $e = 0$. Let $m \in S_d$. Then $m \in A$ so $0 \leq L(m)$. On the other hand, by Lemma 22 also $1 - m \in A$, so $0 \leq L(1-m) = L(1) - L(m) = -L(m)$. Thus $L(m) = 0$ for all $m \in S_d$, contradicting the nontriviality of L.

We claim that $E(p) := L(p)/L(1)$ is a $1/R$-pseudoexpectation. By construction E is linear and satisfies conditions 1 and 2 of Definition 20. It remains to show conditions 3 and 4. We have that $E(p) \geq e/L(1) \geq -1$ for $p \in A$ and $E(p) \leq e/L(1) \leq 0$ for $p \in B$.

For condition 3, let $m \in S_d$. Then $Rm \in A$ so $E(Rm) \geq -1$ hence $E(m) \geq -1/R$. For condition 4, we must show $|E(mq)| \leq 1/R$ for any $m \in S_d$ and $q \in Q$. We have $-1 \pm Rmq \in B$ so $E(-1 \pm Rmq) \leq 0$, where we are using $\pm$ to mean that the inequalities hold whether we write $+$ or $-$. Thus $-1 \leq \pm RE(mq)$ giving $|E(mq)| \leq 1/R$ as required. ◀

We can now use our semantic characterization of bounded magnitude, degree-d SA to prove feasible disjunction, similar to the proof of Theorem 18. We first need a technical lemma.

▶ **Lemma 24.** *If E is an ε-pseudoexpectation for Q over degree d, then $E(m) \leq 1 + d\varepsilon$ for every $m \in S_d$.*

Proof. Let v be the sum of d monomials such that $1 - m \equiv v$ given by Lemma 22. Then $E(1-m) = E(v) \geq -d\varepsilon$, and then we can use linearity. ◀

▶ **Theorem 25.** *Let $Q(x)$ and $P(y)$ be systems of polynomials in disjoint sets of variables x, y and let $R \geq d$. Suppose that neither Q nor P has an R-bounded SA refutation of degree d. Then $Q \wedge P$ has no $R/2$-bounded refutation of degree d.*

Proof. By the assumption on P and Q and the completeness lemma, there are $1/R$-pseudoexpectations E_x and E_y for respectively P and Q (in their respective variables), both over degree d.

Let S be the set of monomials of degree at most d in x-variables and also of degree at most d in y-variables. We define $E : \mathbb{R}[S] \to \mathbb{R}$ on monomials $m_x m_y \in S$ by $E(m_x m_y) = E_x(m_x)E_y(m_y)$ and extend by linearity to the whole space. Note that for any polynomials $p = \sum_i m_i$ in only x-variables and $q = \sum_j n_j$ in only y-variables we have

$$\begin{aligned} E(pq) = E(\sum_i m_i \sum_j n_j) &= \sum_i \sum_j E(m_i n_j) \\ &= \sum_i \sum_j E_x(m_i)E_y(n_j) = E_x(p)E_y(q). \end{aligned}$$

We claim that conditions 1–4 of Definition 20 hold for E, with $\varepsilon = 2/R$, which gives the result by the soundness lemma.

Condition 1 is clear. For condition 2, suppose $p \equiv q$ modulo the ideal generated by the Boolean and negation axioms for both x and y variables. Then $p - q$ can be written as a sum of terms of the form $a_{x_i} b_x c_y$ or $a_{y_i} b_x c_y$, where a_{x_i} and a_{y_i} are axioms from the ideal for a single x or y variable, and b_x and c_y are polynomials in only x or only y variables. But $E_x(a_{x_i} b_x) = 0$ and $E_y(a_{y_i} c_y) = 0$ for terms of this form, so $E(p - q) = 0$.

For condition 3, let $m_x m_y$ be a product of an x-monomial and a y-monomial. We have $-1/R \le E_x(m_x), E_y(m_y) \le 2$, where the upper bound is from Lemma 24 and the assumption that $R \ge d$. Thus $E(m_x m_y) = E_x(m_x)E_y(m_y) \ge -2/R$.

For condition 4 we must show $|E(mq)| \le 2/R$ for any $m = m_x m_y$ and any $q \in P \cup Q$. Suppose without loss of generality that $q \in Q$, so only has y-variables. Then $|E(mq)| = |E_x(m_x)||E_y(m_y q_y)|$. As above, by Lemma 24 we have $|E_x(m_x)| \le 2$, and $|E_y(m_y q_y)| \le 1/R$ by condition 4 for E_y. ◀

As an immediate corollary we get

▶ **Corollary 26.** SA *has μ^*_{SA}-feasible disjunction.*

3.3 Two Failures of feasible Disjunction

In this section, we consider feasible disjunction for size, using the standard definition of feasible disjunction. That is, for every pair F, G of CNFs over disjoint sets of variables, if there is a refutation π of $F \wedge G$ then there is a refutation either of F or of G of size polynomial in the size of π. We show that this fails for RevRes and RevResT.

Let ϕ_n be the CNF $\mathrm{SoD}_n \circ \oplus$, that is, the sink-of-DAG principle lifted by replacing each variable with an XOR of size 2. The following proposition is based on a remark in [22].

▶ **Proposition 27.** *The family ϕ_n has quasipolynomial size, polylog width resolution refutations. On the other hand, it requires super-quasipolynomial size refutations in* uSA.

Proof. The upper bound is essentially by the robustness of polylog-width resolution. Let $F(x_1, ..., x_n)$ be a k-CNF with m clauses and n variables. Then $(F \circ \oplus)(y_1, z_1, ..., y_n, z_n)$ is a $2k$-CNF with at most $m2^k$ clauses (each x_i is substituted by $y_i \oplus z_i$). Note that every clause in F turns into a $2k$-CNF in $F \circ \oplus$ with at most 2^k clauses. So if k is polylogarthmic in n, it is straightforward to see that $\mathsf{Search}(F \circ \oplus) \le \mathsf{Search}(F)$. This means that $\mathsf{Search}(\phi_n) \le \mathsf{Search}(\mathrm{SoD}_n)$ which implies that $\mathsf{Search}(\phi_n) \in \mathsf{PLS}$ by Proposition 15. Therefore the family ϕ_n has quasipolynomial size, polylog width resolution refutations by Proposition 13 and the fact that there are at most $n^{\mathsf{poly}(\log n)}$ many clauses of width $\mathsf{poly}(\log n)$.

For the lower bound, we follow the proof sketched in [22] (although we do not get as strong a bound as claimed there). We adapt the argument used for resolution in Theorem 4.2 of [3]. Let π be a uSA refutation of ϕ_n. We apply the following random restriction ρ. Independently for each i, select one of the pair (y_i, z_i) and set it to either 0 or 1, making each choice uniformly at random. Note that $\phi_n \restriction_\rho$ is equivalent to SoD_n up to renaming of variables and literals. Let m be a monomial in π. We may assume that each variable appears in m at most once. Then $\mathbf{Pr}[m\restriction_\rho \neq 0] \le (\frac{3}{4})^{\mathsf{deg}(m)}$. The reason is that if one of y_i or z_i appears in m (as a positive or negative literal), then this literal is set to 0 with probability $\frac{1}{4}$. If both variables appear in m, then the probability that the product of these two literals is set to 0 is $\frac{1}{2} < (\frac{3}{4})^2$.

Thus for any d, the expected number of monomials in π of degree at least d which remain in $\pi\restriction_\rho$ is at most $(\frac{3}{4})^d$ times the number of monomials in π, by linearity of expectation. Let $d := n^{\frac{1}{\log\log n}}$. Now suppose for a contradiction that the size of π is less than $(\frac{4}{3})^d$. Then there must exist a restriction ρ such that $\pi\restriction_\rho$ has degree d. But, since $\phi_n\restriction_\rho$ is essentially SoD_n and $\pi\restriction_\rho$ still has the form of an uSA refutation, this implies that SoD_n has $n^{\frac{1}{\log\log n}}$ degree, $(\frac{4}{3})^{n^{\frac{1}{\log\log n}}}$ size SA refutations, contradicting Proposition 15. ◀

In [28], Theorem 4.1, a family of bipartite graphs G_n is constructed, between $n+1$ pigeons and n holes, with degree bounded by a constant, such that the perfect matching principle $\mathsf{PMP}_{\mathsf{G_n}}$ requires exponential size to refute in resolution. Let ψ_n be $\mathsf{PMP}_{\mathsf{G_n}}$, which we could also call $\mathsf{onto\text{-}FPHPG_n}$, that is, the onto functional pigeonhole principle on G_n.

▶ **Proposition 28.** *The family ψ_n has polynomial sized, polylog degree refutations in* uNS*. On the other hand, it requires exponential sized refutations in resolution.*

Proof. The lower bound is by [28]. The upper bound is by a small adaptation of the standard proof of onto functional PHP in an algebraic system [47]. Suppose G_n has degree d, and that $\mathsf{PMP}_{\mathsf{G_n}}$ is written in variables x_e for each edge e in G_n, where an edge is formally a pair of nodes. $\mathsf{PMP}_{\mathsf{G_n}}$ consists of degree d axioms $\prod_{e\ni i}\bar{x}_e$ for each node i and degree 2 axioms $x_e x_f$ for each pair e, f of distinct edges sharing a node. For the uNS refutation, observe that

$$\prod_{e\ni i}\bar{x}_e \equiv \prod_{e\ni i}(1-x_e) = 1 - \sum_{e\ni i} x_e + B_i$$

where B_i is a degree at most d combination of the $x_e x_f$ axioms. Summing over the set P of $n+1$ pigeon nodes and rearranging gives

$$n+1-\sum_{i\in P}\sum_{e\ni i} x_e \equiv \sum_{i\in P}(\prod_{e\ni i}\bar{x}_e - B_i),$$

where the double sum on the left is precisely the sum over all variables, and the right hand side is a low-degree combination of axioms. Summing over holes instead of pigeons gives a similar equivalence, with n in place of $n+1$. Subtracting the second equivalence from the first gives the refutation. ◀

▶ **Proposition 29** ([22])**.** uSA *simulates* RevRes

Proof. It was observed in [22] that the constructions for simulating resolution by SA in the literature [13, 1] also give this simulation. We outline how this goes. Suppose we are given a RevRes refutation of a CNF F. For each multiset $M_i = \{C_1, \dots, C_m\}$ of clauses in the refutation, let p_i be the polynomial $r_1 + \cdots + r_m$, where r_j is the standard translation of the clause C_j into a monomial which is zero if and only if C_j is true (where the monomial may used negated variables). Intuitively the equation $p_i = 0$ says the same thing as "every clause in M_i is true". It is easy to see by the structure of the reversible resolution and reversible weakening rules that $p_i \equiv p_{i+1}$ for each i, and in fact $p_{i+1} - p_i$ is simply a monomial times a negation axiom. Moreover it follows from the other conditions on a RevRes refutation that p_1 is a sum of (translations into monomials of) axioms, and that the last line p_t has the form $1 + v$ where 1 is the translation of the empty clause and v is a sum of monomials with positive coefficients. Thus we have $p_1 \equiv 1 + v$, which we can rearrange as $-1 \equiv v - p_1$, which is the required uSA refutation of F. ◀

We may assume that ϕ_n and ψ_n use disjoint sets of variables. By the next lemma and the fact that RevResT is a subsystem of RevRes we get the failure of feasible disjunction for both systems.

▶ **Lemma 30.**
1. *$\phi_n \wedge \psi_n$ has quasipolynomial sized refutations in* RevResT
2. *ϕ_n requires super-quasipolynomial sized refutations in* RevRes
3. *ψ_n requires exponential sized refutations in* RevRes*.*

Proof. For item 1 we use Theorem 6 of [22], which characterizes the proof measure of a CNF in $\mathsf{RevResT}$ (that is, minimum degree plus log of size over all refutations) as the sum of the measures in resolution and uNS. In particular, a CNF has quasipolynomial size, polylog width refutations in $\mathsf{RevResT}$ if and only if it has such refutations in both resolution and uNS. We have this for $\phi_n \wedge \psi_n$ by the two propositions above.

For item 2, if ϕ_n had small refutations in RevRes then by Proposition 29 it would also have small refutations in uSA, contradicting Proposition 27. Item 3 follows directly from the lower bound for resolution in Proposition 28. ◀

4 Results on intersections

For the purposes of this paper, we define the *classical TFNP classes* to be the six classes PLS, PPA, PPP, PPADS, PPAD, and CLS depicted in Figure 1. Among these classes we have the relations

$$\mathsf{PPP} \supset \mathsf{PPADS} \supset \mathsf{PPAD} \supset \mathsf{CLS}; \quad \mathsf{PPA} \supset \mathsf{PPAD}; \quad \mathsf{PLS} \supset \mathsf{CLS}$$

and no inclusions hold, other than those implied by the above inclusions [2, 41, 22]. This structure does not change if we replace PPA with PPA_q for any prime q, and, furthermore, there is no inclusion between classes PPA_q and PPA_r, for distinct primes [31, 26].

We say that a *nontrivial intersection* is an intersection $A \cap B$ of two classical classes where neither $A \subseteq B$ nor $B \subseteq A$.

▶ **Lemma 31.** *Suppose F_n, G_n are narrow CNF families and A, B are* TFNP *classes with* $\mathsf{Search}(F_n) \in A$ *and* $\mathsf{Search}(G_n) \in B$*. Then* $\mathsf{Search}(F_n \wedge G_n) \in A \cap B$*.*

Proof. Clearly $\mathsf{Search}(F_n \wedge G_n) \leq \mathsf{Search}(F_n)$, since an assignment to the variables of $F_n \wedge G_n$ is in particular an assignment to the variables of F_n, and finding a false clause in F_n gives us a false clause in $F_n \wedge G_n$. Similarly $\mathsf{Search}(F_n \wedge G_n) \leq \mathsf{Search}(G_n)$. ◀

▶ **Theorem 32.** *With the exception of* $\mathsf{PPAD} \cap \mathsf{PLS} = \mathsf{CLS}$*, no nontrivial intersection is equal to a classical class. No two distinct nontrivial intersections are equal to each other.*

Proof. We start by going through each classical class in turn, and showing that intersecting with that class does not lead to any unexpected collapses.

Intersections with PLS. We have

$$\mathsf{PLS} \supset \mathsf{PPP} \cap \mathsf{PLS} \supseteq \mathsf{PPADS} \cap \mathsf{PLS} \supseteq \mathsf{CLS}\ (= \mathsf{PPAD} \cap \mathsf{PLS});$$
$$\mathsf{PLS} \supset \mathsf{PPA} \cap \mathsf{PLS} \supseteq \mathsf{CLS}.$$

Note that the intersection has collapsed PPAD and CLS together. We claim these classes are all distinct. This follows from the following claims.

1. $\mathsf{PPP} \cap \mathsf{PLS} \not\subseteq \mathsf{PPADS}$. Suppose this inclusion held. By Propositions 14 and 15 we have $\mathsf{Search}(\mathrm{BPHP}_n) \in \mathsf{PPP}$ and $\mathsf{Search}(\mathrm{SoD}_n) \in \mathsf{PLS}$. By Lemma 31 it follows that $\mathsf{Search}(\mathrm{BPHP}_n \wedge \mathrm{SoD}_n) \in \mathsf{PPADS}$. By the uSA characterization of PPADS (in Proposition 13) the conjunction $\mathrm{BPHP}_n \wedge \mathrm{SoD}_n$ has uSA refutations of measure $\mu_{\mathsf{uSA}} \leq \mathsf{poly}(\log n)$, and thus in particular has SA derivations of measure $\mu^*_{\mathsf{SA}} \leq \mathsf{poly}(\log n)$. We may assume that BPHP_n and SoD_n have no variables in common, so we can apply μ^*_{SA}-feasible disjunction for SA (Corollary 26) to conclude that either

   a. for infinitely many n, BPHP_n has SA refutations of measure $\mu^*_{\mathsf{SA}} \leq \mathsf{poly}(\log n)$. Hence in particular they have degree $\mathsf{poly}(\log n)$, which is impossible by Proposition 14.
   b. for infinitely many n, SoD_n has SA refutations of measure $\mu^*_{\mathsf{SA}} \leq \mathsf{poly}(\log n)$, that is, simultaneously of degree $\mathsf{poly}(\log n)$ and of magnitude quasipolynomial in n. This is impossible by Proposition 15.

2. $\mathsf{PPADS} \cap \mathsf{PLS} \not\subseteq \mathsf{PPA}$. This item was proved in Lemma 2 of [22]. Here we repeat the argument for the sake of completeness. Suppose this inclusion held. By Lemma 12 and Proposition 15 we have $\mathsf{Search}(\mathsf{CNF}(Q_n)) \in \mathsf{PPADS}$ and $\mathsf{Search}(\mathrm{SoD}_n) \in \mathsf{PLS}$, where Q is any complete problem for PPADS. By Lemma 31 it follows that $\mathsf{Search}(\mathsf{CNF}(Q_n) \wedge \mathrm{SoD}_n) \in \mathsf{PPA}$, and hence this conjunction has $\mathsf{poly}(\log n)$-degree $\mathsf{NS}_{\mathbb{F}_2}$ refutations, by Proposition 13. We may assume $\mathsf{CNF}(Q_n)$ and SoD_n have no variables in common, so we can apply deg-feasible disjunction for $\mathsf{NS}_{\mathbb{F}_2}$ (Theorem 18) to conclude that either
   a. for infinitely many n, SoD_n has $\mathsf{NS}_{\mathbb{F}_2}$ refutation of degree $\mathsf{poly}(\log n)$. This is impossible by Proposition 15.
   b. for all sufficiently large n, $\mathsf{CNF}(Q_n)$ has $\mathsf{NS}_{\mathbb{F}_2}$ refutations of degree $\mathsf{poly}(\log n)$. Then by Proposition 13 and Lemma 12, we have $Q \in \mathsf{PPA}$. This is impossible, as Q is complete for PPADS.

3. $\mathsf{PPA} \cap \mathsf{PLS} \not\subseteq \mathsf{PPA}_3$. This is proved in exactly the same way as the previous item, using $\mathsf{NS}_{\mathbb{F}_3}$ instead of $\mathsf{NS}_{\mathbb{F}_2}$ and taking Q to be any complete problem for PPA.

By items 1 and 2 all inclusions in the first row are strict, where to apply 2 we use that $\mathsf{CLS} \subseteq \mathsf{PPA}$. For $\mathsf{PPA} \cap \mathsf{PLS}$, by item 2 it is distinct from the two classes in the middle of the first row, and by item 3 it is distinct from CLS, since $\mathsf{CLS} \subseteq \mathsf{PPAD} \subseteq \mathsf{PPA}_3$.

Intersections with PPP. We have

$$\mathsf{PPP} \supset \mathsf{PPADS} \supset \mathsf{PPAD} \supset \mathsf{CLS};$$
$$\mathsf{PPP} \supset \mathsf{PLS} \cap \mathsf{PPP} \supseteq \mathsf{CLS};$$
$$\mathsf{PPP} \supset \mathsf{PPA} \cap \mathsf{PPP} \supseteq \mathsf{PPAD}$$

where PPADS, PPAD and CLS are unchanged by intersection with PPP. We claim the classes above are all distinct. This follows for the classes in the first two rows by item 1 above. For $\mathsf{PPA} \cap \mathsf{PPP}$ we have

4. $\mathsf{PPA} \cap \mathsf{PPP} \not\subseteq \mathsf{PPADS}$. Suppose this inclusion held. Then, using a similar argument to item 2, $\mathsf{CNF}(Q_n) \wedge \mathrm{BPHP}_n$ has SA refutations of measure $\mu^*_{\mathsf{SA}} \leq \mathsf{poly}(\log n)$, where Q is any complete problem for PPA. We then get a contradiction from μ^*_{SA}-feasible disjunction for SA, since BPHP_n requires large SA-degree for all large n and $Q \notin \mathsf{PPADS}$.

Thus $\mathsf{PPA} \cap \mathsf{PPP}$ is different from anything in the top row. The remaining possible equality is $\mathsf{PPA} \cap \mathsf{PPP} = \mathsf{PLS} \cap \mathsf{PPP}$, but this is impossible, as PPAD is included in the left hand side but not in PLS.

Intersections with PPA. We have

$$\mathsf{PPA} \supset \mathsf{PPAD} \supset \mathsf{CLS};$$
$$\mathsf{PPA} \supset \mathsf{PPP} \cap \mathsf{PPA} \supseteq \mathsf{PPADS} \cap \mathsf{PPA} \supseteq \mathsf{PPAD};$$
$$\mathsf{PPA} \supset \mathsf{PLS} \cap \mathsf{PPA} \supseteq \mathsf{CLS}$$

where PPAD and CLS are unchanged by intersection with PPA. We claim the classes above are all distinct. We have

5. $\mathsf{PPADS} \cap \mathsf{PPA} \not\subseteq \mathsf{PPA}_3$. Suppose this inclusion held. We get that $\mathsf{CNF}(Q_n) \wedge \mathrm{Count}_n^2$ has $\mathsf{poly}(\log n)$-degree $\mathsf{NS}_{\mathbb{F}_3}$ refutations, where Q is complete for PPADS. We then proceed as in item 2, using deg-feasible disjunction for $\mathsf{NS}_{\mathbb{F}_3}$ and the degree lower bounds for $\mathsf{NS}_{\mathbb{F}_3}$ from Proposition 16.

Item 5, together with item 4, imply that the first two rows are all distinct. For $\mathsf{PLS} \cap \mathsf{PPA}$, by item 3 it is different from anything in the first row, and since $\mathsf{PPAD} \not\subseteq \mathsf{PLS}$ it is different from anything in the second row.

Intersections with PPADS. We have

$$\begin{aligned} &\mathsf{PPADS} \supset \mathsf{PPAD} \supset \mathsf{CLS};\\ &\mathsf{PPADS} \supset \mathsf{PPA} \cap \mathsf{PPADS} \supseteq \mathsf{PPAD};\\ &\mathsf{PPADS} \supset \mathsf{PLS} \cap \mathsf{PPADS} \supseteq \mathsf{CLS} \end{aligned}$$

where PPAD and CLS are unchanged by the intersection, and PPP collapses to PPADS. We claim the classes above are all distinct. $\mathsf{PPA} \cap \mathsf{PPADS}$ is distinct from everything in the first row by item 5. Then $\mathsf{PLS} \cap \mathsf{PPADS}$ is distinct from everything else by item 2.

Intersections with PPAD. We have

$$\mathsf{PPAD} \supset \mathsf{CLS}.$$

Under the intersection, everything above PPAD collapses to PPAD, and PLS collapses to CLS. So there is nothing to show.

We have now shown that no nontrivial intersection is equal to a classical class. It remains to show that no two distinct nontrivial intersections are equal to each other. Suppose for a contradiction that we have such an equality $A \cap B = C \cap D$. We have dealt with all cases where one class appears on both sides, so we may assume the classes A, B, C, D are all distinct. None of them can be CLS, by the condition that the intersections are nontrivial. If we consider the four classes $\mathsf{PPP}, \mathsf{PPADS}, \mathsf{PPAD}, \mathsf{PPA}$, it is similarly impossible to form two nontrivial intersections from these, since PPAD is contained in the other three classes. So without loss of generality we may assume $A = \mathsf{PLS}$ and thus $C \cap D \subseteq \mathsf{PLS}$. But this is impossible, as for all remaining choices of C and D we have $\mathsf{PPAD} \subseteq C \cap D$. ◀

▶ **Theorem 33.** *Theorem 32 remains true if we add $\{\mathsf{PPA}_q : \text{ prime } q > 2\}$ to the set of classical classes.*

Proof. Everything in the proof of Theorem 32 goes through if we replace PPA with PPA_q – the only changes we need to make are replacing $\mathsf{NS}_{\mathbb{F}_2}$ with $\mathsf{NS}_{\mathbb{F}_q}$ and, if $q = 3$, replacing $\mathsf{NS}_{\mathbb{F}_3}$ with $\mathsf{NS}_{\mathbb{F}_2}$ (or with $\mathsf{NS}_{\mathbb{F}_r}$ for any prime r different from 3); all the relevant upper and lower bounds still work.

This leaves equalities involving at least two classes PPA_q, PPA_r, with q, r distinct primes. We consider the possible forms these could take.

The form $\mathsf{PPA}_q \cap B = \mathsf{PPA}_r$. This implies $\mathsf{PPA}_r \subseteq \mathsf{PPA}_q$, which is false.

The form $\mathsf{PPA}_q \cap \mathsf{PPA}_r = C$. By the previous case, C must be a classical class, so C can only be PPAD (since we know $\mathsf{PPAD} \subseteq \mathsf{PPA}_q \cap \mathsf{PPA}_r$). But by Section 7.2 of [23], this is not the case.

The form $\mathsf{PPA}_q \cap B = \mathsf{PPA}_r \cap D$. By feasible disjunction for $\mathsf{NS}_{\mathbb{F}_q}$, for each n we get a low-degree $\mathsf{NS}_{\mathbb{F}_q}$ refutation of either Count_n^r or $\mathsf{CNF}(D_n)$. Since the first of these is impossible for large n, it must asymptotically be a refutation of $\mathsf{CNF}(D_n)$, implying $D \subseteq \mathsf{PPA}_q$. Thus D must be either PPAD or CLS, contradicting that the right hand side was a nontrivial intersection.

The form $\mathsf{PPA}_q \cap \mathsf{PPA}_r = C \cap D$. By the above case neither C nor D can be PPA_s for a prime s. Since $C \cap D \supseteq \mathsf{PPAD}$ and $\mathsf{PLS} \not\subseteq \mathsf{PPAD}$, we must have that C and D are two classes from the list $\mathsf{PPP}, \mathsf{PPADS}, \mathsf{PPAD}$; but in this case $C \cap D$ is a trivial intersection. ◀

5 A Combinatorial Characterization of PPADS ∩ PPA

In this section, we give a combinatorial characterization of the intersection of PPADS and PPA. Unlike the separations for the decision tree variants of TFNP classes established in the previous section, the characterization holds also in the standard "white-box" model of TFNP. As discussed in Section 1.1.3, the search problem Leaf-Or-Sink is to find a leaf (i.e., a vertex with a single neighbor) distinct from 0^n or a sink in a directed graph where each vertex is of degree at most two and 0^n is a source of degree one.

Next, we give the formal definition of the problem. To avoid cumbersome notation, we first define graph induced by a triple of Boolean circuits.

▶ **Definition 34.** *Given Boolean circuits $N_0, N_1 \colon \{0,1\}^n \to \{0,1\}^n$, and $D \colon \{0,1\}^{2n} \to \{0,1\}$, we define a directed graph $G_{N_0,N_1,D} = (V,E)$, where $V = \{0,1\}^n$ and, for $u, v \in \{0,1\}^n$, $(u,v) \in E$ if and only if $u \neq v$, $u \in \{N_0(v)\} \cup \{N_1(v)\}$, $v \in \{N_0(u)\} \cup \{N_1(u)\}$, $D(u,v) = D(v,u)$, and either*

1. *$u < v$ and $D(u,v) = 1$, or*
2. *$u > v$ and $D(u,v) = 0$.*

Note that the above definition induces a graph on $\{0,1\}^n$, where each vertex is of degree at most two and, for any pair of vertices u and v, there is at most one directed edge adjacent to u and v.

▶ **Definition 35** (Leaf-Or-Sink). *The search problem Leaf-Or-Sink is defined by the relation:*

Instance: *Boolean circuits $N_0, N_1 \colon \{0,1\}^n \to \{0,1\}^n$, and $D \colon \{0,1\}^{2n} \to \{0,1\}$ such that, in the corresponding graph $G_{N_0,N_1,D}$, the vertex 0^n is a source of degree one.*

Solution: *Either a vertex of degree one distinct from 0^n or any sink in $G_{N_0,N_1,D}$.*

Next, we show that Leaf-Or-Sink is complete for PPADS ∩ PPA. The containment in PPADS ∩ PPA is trivial. The high-level overview of our proof of PPADS ∩ PPA-hardness is provided in Section 1.1.3.

Below, we recall the formal definitions of the PPA-complete problem Leaf and the PPADS-complete problem Sink. Similarly to the above definition of Leaf-Or-Sink, we phrase the definition in terms of the graph induced by the Boolean circuit(s) given as input to the problem. For a Leaf instance $N \colon \{0,1\}^n \to \{0,1\}^{2n}$, G_N is a graph on $\{0,1\}^n$ and there is an undirected edge between u and v iff $u \neq v$, $u \in N(v)$ and $v \in N(u)$. Similarly, for a Sink instance $S, P \colon \{0,1\}^n \to \{0,1\}^n$, $G_{S,P}$ is a graph on $\{0,1\}^n$ and there is a directed edge from u to v iff $u \neq v$, $S(u) = v$ and $P(v) = u$.

▶ **Definition 36** (Leaf). *The search problem Leaf is defined by the relation:*

Instance: *Boolean circuit $N\colon \{0,1\}^n \to \{0,1\}^{2n}$ such that, in the corresponding undirected graph G_N, the vertex 0^n is a leaf.*

Solution: *A leaf in G_N distinct from 0^n.*

▶ **Definition 37** (Sink). *The search problem Sink is defined by the relation:*

Instance: *Boolean circuits $S, P\colon \{0,1\}^n \to \{0,1\}^n$ such that, in the corresponding directed graph $G_{S,P}$, the vertex 0^n is a source of degree one.*

Solution: *A sink in $G_{S,P}$.*

Finally, we state and prove the main theorem of this section.

▶ **Theorem 38.** *Leaf-Or-Sink is complete for* PPADS ∩ PPA.

Proof. Any instance (N_0, N_1, D) of Leaf-Or-Sink can be reduced to both the PPA-complete problem Leaf and the PPADS-complete problem Sink-Of-Line. First, the circuits N_0 and N_1 specify an undirected Leaf graph where all the solutions, i.e., leaves distinct from 0^n, correspond either to a source of degree one distinct from 0^n or a sink (of degree one) in $G_{N_0,N_1,D}$, i.e., to a solution of the Leaf-Or-Sink instance (N_0, N_1, D). Second, the directed graph $G_{N_0,N_1,D}$ can be locally transformed into a directed graph satisfying the stronger requirement that any vertex $v \in \{0,1\}^n$ is of in-degree and out-degree at most one: for any vertex of in-degree two or out-degree two, simply erase the adjacent edges. The resulting graph is an instance of Sink-Of-Line where all the solutions, i.e., sinks, are either sinks or neighbors of a sink in $G_{N_0,N_1,D}$ and, thus, can be used to efficiently find a solution to the Leaf-Or-Sink instance (N_0, N_1, D). Therefore, Leaf-Or-Sink is contained in PPADS ∩ PPA.[2]

Next, we show that Leaf-Or-Sink is PPADS ∩ PPA-hard, i.e., we give a reduction from a PPADS ∩ PPA-complete problem to Leaf-Or-Sink. Consider an arbitrary problem in PPADS ∩ PPA given by an instance (S, P) of Sink and instance N of Leaf. For ease of exposition, suppose that all the circuits S, P, and N take n-bit inputs (the general case can be handled easily by padding to the same input length).

Using the instances (S, P) and N, we define an instance of Leaf-Or-Sink on $2n$ bits in the following phases, starting with only isolated vertices:

1. Put an undirected edge between vertices $(u, 0^n)$ and $(v, 0^n)$ if there is an edge between u and v in the undirected graph induced by the Leaf instance N.
2. Orient the edges in the graph w.r.t. the lexicographic order on $\{0,1\}^n$.
3. Erase both edges adjacent to any sink $(w, 0^n)$ of in-degree two and, for any neighbor u of w in the Leaf instance N, embed the instance (S, P) of Sink-Of-Line on the vertices of the form $(u, \cdot)$. Formally, add the edge $((u, x), (u, y))$ iff (x, y) is an edge in the directed graph induced by (S, P). For any source $s \neq 0^n$ in the graph induced by (S, P) and neighbors $(u, 0^n)$ and $(v, 0^n)$ of any sink $(w, 0^n)$ of degree two, add edges $((w, s), (u, s))$ and $((w, s), (v, s))$.

[2] In more detail, we need to provide an implicit representation of the PPADS instance via a successor and predecessor circuit. Note that these can be implemented via a constant number of queries to the circuits (N_0, N_1, D).

Note that we can assume without loss of generality that, in the graph constructed after the second phase, any sink of in-degree two is connected only to vertices of in-degree and out-degree at most one.[3] In particular, no pair of sinks of in-degree two share a predecessor and, therefore, the steps of the third phase are well defined as each embedding performed in the third phase involves distinct vertices.

The resulting graph is clearly an instance of Leaf-Or-Sink. Importantly, neither the new sources of degree two of the form (w, s) nor the balanced vertices (u, s) and (v, s) of in-degree and out-degree one added during the third phase constitute a solution to Leaf-Or-Sink. Let (u, x) be a sink in the resulting graph. By our construction, either $x = 0^n$ and $u \neq 0^n$ is a leaf in the Leaf graph induced by N or x is a sink in the Sink-Of-Line graph induced by (S, P). Let (u, x) be a source in this graph of degree one. By our construction, it must be the case that $x = 0^n$ and $u \neq 0^n$ is a leaf in the Leaf graph induced by N or x is a sink in the Sink-Of-Line graph induced by (S, P). ◀

Note that, by allowing fewer or more types of vertices as solutions in the definition of Leaf-Or-Sink, we can provide alternative formulations for the problems defining PPA, PPADS, and PPAD. The variant, where we search only for leaves distinct from the distinguished source is PPA-complete. The variant, where we search for any sink or source (of degree one or two) distinct from the distinguished source is PPAD-complete. The variant where we search only for sinks is PPADS-complete.

References

1. Albert Atserias, Massimo Lauria, and Jakob Nordström. Narrow proofs may be maximally long. *ACM Trans. Comput. Log.*, 17(3):19, 2016. doi:10.1145/2898435.
2. Paul Beame, Stephen A. Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of NP search problems. *J. Comput. Syst. Sci.*, 57(1):3–19, 1998. doi:10.1006/jcss.1998.1575.
3. Eli Ben-Sasson. Size-space tradeoffs for resolution. *SIAM Journal on Computing*, 38(6):2511–2525, 2009.
4. Josh Buresh-Oppenheim and Tsuyoshi Morioka. Relativized NP search problems and propositional proof systems. In *19th Annual IEEE Conference on Computational Complexity (CCC 2004), 21-24 June 2004, Amherst, MA, USA*, pages 54–67. IEEE Computer Society, 2004. doi:10.1109/CCC.2004.1313795.
5. Joshua Buresh-Oppenheim. On the TFNP complexity of factoring. https://www.cs.toronto.edu/~bureshop/factor.ps, 2006.
6. Samuel R. Buss. Lower bounds on Nullstellensatz proofs via designs. In *Proof complexity and feasible arithmetics*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pages 59–71. American Mathematical Society, Providence, RI, USA, 1998.
7. Samuel R. Buss, Russell Impagliazzo, Jan Krajíček, Pavel Pudlák, Alexander A. Razborov, and Jiri Sgall. Proof complexity in algebraic systems and bounded depth Frege systems with modular counting. *Computational Complexity*, 6:256–298, 1996.
8. Samuel R. Buss, Leszek Aleksander Kołodziejczyk, and Neil Thapen. Fragments of approximate counting. *The Journal of Symbolic Logic*, 79(2):496–525, 2014.
9. Samuel R. Buss and Jan Krajíček. An application of boolean complexity to separation problems in bounded arithmetic. *Proceedings of the London Mathematical Society*, 3(1):1–21, 1994.

[3] This property could be achieved by a simple additional transformation. For every edge $(u, v) \in E$, we can introduce a new vertex $w_{u,v}$ and add the edges $(u, w_{u,v})$ and $(w_{u,v}, v)$. This transformation is locally computable, preserves the structure of the graph, and ensures the claimed structure w.r.t. degrees of vertices.

10 Samuel R. Buss and Grigori Mints. The complexity of the disjunction and existential properties in intuitionistic logic. *Annals of Pure and Applied Logic*, 99(1-3):93–104, 1999.

11 Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo. Using the groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 174–183, 1996.

12 Stefan Dantchev, Nicola Galesi, Abdul Ghani, and Barnaby Martin. Proof complexity and the binary encoding of combinatorial principles. *arXiv preprint arXiv:2008.02138*, 2020.

13 Stefan Dantchev, Barnaby Martin, and Mark Rhodes. Tight rank lower bounds for the Sherali–Adams proof system. *Theoretical Computer Science*, 410(21-23):2054–2063, 2009.

14 Constantinos Daskalakis and Christos H. Papadimitriou. Continuous local search. In Dana Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 790–804. SIAM, 2011. doi:10.1137/1.9781611973082.62.

15 Constantinos Daskalakis, Christos Tzamos, and Manolis Zampetakis. A converse to Banach's fixed point theorem and its CLS-completeness. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 44–50. ACM, 2018. doi:10.1145/3188745.3188968.

16 Ben Davis and Robert Robere. Colourful TFNP and propositional proofs. In Amnon Ta-Shma, editor, *38th Computational Complexity Conference, CCC 2023, July 17-20, 2023, Warwick, UK*, volume 264 of *LIPIcs*, pages 36:1–36:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.CCC.2023.36.

17 Susanna de Rezende, Aaron Potechin, and Kilian Risse. Clique is hard on average for unary Sherali-Adams. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, 2023. To appear.

18 John Fearnley, Paul Goldberg, Alexandros Hollender, and Rahul Savani. The complexity of gradient descent: CLS = PPAD $\cap$ PLS. *J. ACM*, 70(1):7:1–7:74, 2023. doi:10.1145/3568163.

19 Michal Garlík. Failure of feasible disjunction property for k-DNF resolution and NP-hardness of automating it. *CoRR*, abs/2003.10230, 2020. arXiv:2003.10230.

20 M. Göös, P. Kamath, R. Robere, and D. Sokolov. Adventures in Monotone Complexity and TFNP. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 38:1–38:19, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ITCS.2019.38.

21 Mika Göös, Alexandros Hollender, Siddhartha Jain, Gilbert Maystre, William Pires, Robert Robere, and Ran Tao. Further collapses in TFNP. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*, volume 234 of *LIPIcs*, pages 33:1–33:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CCC.2022.33.

22 Mika Göös, Alexandros Hollender, Siddhartha Jain, Gilbert Maystre, William Pires, Robert Robere, and Ran Tao. Separations in proof complexity and TFNP. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 1150–1161. IEEE, 2022. doi:10.1109/FOCS54457.2022.00111.

23 Mika Göös, Pritish Kamath, Katerina Sotiraki, and Manolis Zampetakis. On the complexity of modulo-q arguments and the Chevalley-Warning theorem. In *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169, pages 19:1–19:42, 2020.

24 Tuomas Hakoniemi. Feasible Interpolation for Polynomial Calculus and Sums-Of-Squares. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*, volume 168 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 63:1–63:14, 2020.

25 Tuomas Hakoniemi. *Size bounds for algebraic and semialgebraic proof systems*. PhD thesis, Universitat Politècnica de Catalunya, 2022.

26 Alexandros Hollender. The classes PPA-k: Existence from arguments modulo k. *Theor. Comput. Sci.*, 885:15–29, 2021. doi:10.1016/j.tcs.2021.06.016.

27 Takashi Ishizuka. The complexity of the parity argument with potential. *J. Comput. Syst. Sci.*, 120:14–41, 2021. doi:10.1016/j.jcss.2021.03.004.

28 Dmitry Itsykson, Vsevolod Oparin, Mikhail Slabodkin, and Dmitry Sokolov. Tight lower bounds on the resolution complexity of perfect matching principles. *Fundamenta Informaticae*, 145(3):229–242, 2016.

29 Emil Jeřábek. Frege systems for extensible modal logics. *Annals of Pure and Applied Logic*, 142(1-3):366–379, 2006.

30 Emil Jeřábek. Integer factoring and modular square roots. *J. Comput. Syst. Sci.*, 82(2):380–394, 2016. doi:10.1016/j.jcss.2015.08.001.

31 Alan Johnson. *Reductions and propositional proofs for total NP search problems*. PhD thesis, University of California, San Diego, 2011.

32 David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? (extended abstract). In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 39–42, 1985.

33 Pritish Kamath. *Some hardness escalation results in computational complexity theory*. PhD thesis, Massachusetts Institute of Technology, 2019.

34 Leszek Aleksander Kolodziejczyk, Phuong Nguyen, and Neil Thapen. The provably total NP search problems of weak second order bounded arithmetic. *Ann. Pure Appl. Log.*, 162(6):419–446, 2011. doi:10.1016/j.apal.2010.12.002.

35 Leszek Aleksander Kolodziejczyk and Neil Thapen. Approximate counting and NP search problems. *J. Math. Log.*, 22(3):2250012:1–2250012:31, 2022. doi:10.1142/S021906132250012X.

36 Jan Krajíček, Alan Skelley, and Neil Thapen. NP search problems in low fragments of bounded arithmetic. *J. Symb. Log.*, 72(2):649–672, 2007. doi:10.2178/jsl/1185803628.

37 J. Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *The Journal of Symbolic Logic*, 62(2):457–486, 1997.

38 Jan Krajíček. *Bounded arithmetic, propositional logic, and complexity theory*, volume 60 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, 1995.

39 Yuhao Li, William Pires, and Robert Robere. Intersection classes in TFNP and proof complexity. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, January 30 - February 2, 2024, University of California, Berkeley, California, USA*, LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. To appear.

40 Nimrod Megiddo and Christos H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theor. Comput. Sci.*, 81(2):317–324, 1991. doi:10.1016/0304-3975(91)90200-L.

41 Tsuyoshi Morioka. Classification of search problems and their definability in bounded arithmetic. *Electron. Colloquium Comput. Complex.*, TR01-082, 2001. arXiv:TR01-082.

42 Tsuyoshi Morioka. The relative complexity of local search heuristics and the iteration principle. *Electron. Colloquium Comput. Complex.*, TR03-051, 2003. arXiv:TR03-051.

43 Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 48(3):498–532, 1994. doi:10.1016/S0022-0000(05)80063-7.

44 Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *J. Symb. Log.*, 62(3):981–998, 1997. doi:10.2307/2275583.

45 Pavel Pudlák. On the complexity of the propositional calculus. In S. Barry Cooper and John K. Truss, editors, *Sets and Proofs*, London Mathematical Society Lecture Note Series, pages 197–218. Cambridge University Press, 1999.

46 Pavel Pudlák. On reducibility and symmetry of disjoint NP pairs. *Theor. Comput. Sci.*, 295:323–339, 2003. doi:10.1016/S0304-3975(02)00411-5.

47 Søren Riis. *Independence in Bounded Arithmetic*. PhD thesis, Oxford University, 1993.

48 Steven Rudich. Super-bits, demi-bits, and NP/qpoly-natural proofs. In José D. P. Rolim, editor, *Randomization and Approximation Techniques in Computer Science, International Workshop, RANDOM'97*, volume 1269 of *Lecture Notes in Computer Science*, pages 85–93. Springer, 1997. doi:10.1007/3-540-63248-4_8.

49 Alan Skelley and Neil Thapen. The provably total search problems of bounded arithmetic. *Proceedings of the London Mathematical Society*, 103(1):106–138, 2011.