Hardness Amplification in Proof Complexity

Paul Beame[*]
Computer Science &
Engineering
University of Washington
Seattle, WA 98195-2350
beame@cs.washington.edu

Trinh Huynh[*†]
Computer Science &
Engineering
University of Washington
Seattle, WA 98195-2350
trinh@cs.washington.edu

Toniann Pitassi[‡]
Computer Science
University of Toronto
Toronto, ON M5S 1A4
toni@cs.toronto.edu

ABSTRACT

We present a general method for converting any family of unsatisfiable CNF formulas that is hard for one of the simplest proof systems – tree resolution – into formulas that require large rank in very strong proof systems, including any proof system that manipulates polynomials of degree at most k (known as Th(k) proofs). These include high degree versions of Lovász-Schrijver and Cutting Planes proofs.

We introduce two very general families of these proof systems, denoted $T^{cc}(k)$ and $R^{cc}(k)$. The proof lines of $T^{cc}(k)$ are arbitrary Boolean functions, each of which can be evaluated by an efficient k-party randomized communication protocol. $T^{cc}(k)$ proofs include Th($k-1$) proofs as a special case. $R^{cc}(k)$ proofs generalize $T^{cc}(k)$ proofs and require only that each inference be checkable by a short k-party protocol.

For all $k \in O(\log \log n)$, our main results are as follows: First, any unsatisfiable CNF formula of high resolution rank can be efficiently transformed into another CNF formula requiring high rank in all $R^{cc}(k)$ systems, and exponential tree size in all $T^{cc}(k)$ systems. Secondly, there are strict rank hierarchies for all $R^{cc}(k)$ systems, and strict tree-size hierarchies for all $T^{cc}(k)$ systems. Finally, we apply our general method to give optimal integrality gaps for low rank $R^{cc}(2)$ proofs for MAX-2t-SAT, which imply optimal integrality gaps for low rank Cutting Planes and Th(1) proofs.

Categories and Subject Descriptors

F.0 [**Theory of Computation**]: General

General Terms

Theory

Keywords

Proof Complexity, Communication Complexity

1. INTRODUCTION

Over the last decade or so there have been a large number of results proving lower bounds on the rank required to refute (or approximately optimize over) systems of constraints in a wide variety of semi-algebraic (a.k.a. polynomial threshold) proof systems. These include systems such as Lovász-Schrijver [25], Cutting Planes [14, 9], Positivstellensatz [15], Sherali-Adams [33], and Lasserre [23] proofs.

Highlights of this work include recent linear rank lower bounds for Lasserre proofs [30, 36] for many constraint optimization problems as well as rank lower bounds for semi-algebraic proof systems for other important optimization problems [7, 31, 13]. In addition to these rank lower bounds, there are a few superpolynomial lower bounds on the size of tree-like proofs in specific semi-algebraic proof systems [16, 15, 20]. Exciting and important as these results are, their proofs rely on delicate constructions of problem-specific local distributions on inputs that satisfy constraints based on the specific rules for each proof system. Furthermore, because there are few effective reductions for such proof systems, lower bounds for one problem usually do not translate to other problems.

A very different approach for proving lower bounds for semi-algebraic proofs was developed in [2], whereby the problems of lower-bounding the rank or tree-like proof size are reduced to a lower bound problem in communication complexity. This allows the results to be applicable to a much wider class of proof systems, called Th(k) proofs, which generalizes the semi-algebraic proof systems discussed above. In these systems, a proof consists of a sequence of lines, each of which is a multivariate polynomial inequality of degree at most k; the only requirement is that each line either expresses an input constraint or is a semantic consequence of a constant number (say two or three) of its predecessors. [2] showed that if an unsatisfiable CNF formula G has a small-rank (or small tree-like size) Th($k-1$) refutation then, over every partition of the variables of G for k parties, there is an efficient k-party randomized NOF protocol that outputs a falsified clause in G. Thus to lower bound the rank of Th($k-1$) proofs it suffices find an unsatisfiable family of CNF formulas with the property that the k-player NOF complexity of this underlying search problem (outputting a falsified clause) is hard.

Though this communication complexity approach was decoupled from the specifics of the proof system, the reduction given in [2] was very problem-specific and delicate. One source of the difficulty was that the clause search problem needs to be hard for randomized protocols to solve but is

[*] Research supported by NSF grant CCF-0830626
[†] Also known as Dang-Trinh Huynh-Ngoc
[‡] Research Supported by NSERC

STOC'10, June 5–8, 2010, Cambridge, Massachusetts, USA.

always easy nondeterministically, as the players can simply guess and verify a violated clause. Much of the delicacy of the argument was in carefully embedding a specific candidate function (set disjointness), which appeared to have these characteristics, into the search problem of an unsatisfiable CNF.

Indeed, using a long and involved argument, [2] showed the feasibility of this communication complexity approach by constructing a particular family of CNF formulas, $(k-1)$-fold Tseitin tautologies over $\Theta(\log n)$-degree LPS expander graphs, such that lower bounds on the k-party randomized NOF communication complexity of the k-party set disjointness function yield rank and tree-like size lower bounds for $\mathrm{Th}(k-1)$ refutations. The recent lower bounds of Lee and Shraibman [24] and Chattopadhyay and Ada [8] for the k-party randomized communication complexity of set disjointness thus yield unconditional rank bounds for $\mathrm{Th}(k)$ proofs. Unfortunately, though the set disjointness bounds apply for k up to $(1-o(1))\log\log n$, the details of the reduction in [2], which was claimed for each constant k, only apply for $k = O(\log\log\log n)$. Moreover, the method only applies to this one particular family of unsatisfiable formulas, and no other lower bounds for $\mathrm{Th}(k)$ proofs have been known by any other method.

In this paper for the first time we provide a simple and general method that produces unsatisfiable formulas requiring proofs of large rank and tree-like size in semi-algebraic proof systems. This applies to a broad range of systems including all of $\mathrm{Th}(k)$ for k up to $(1-o(1))\log\log n$. Our method allows one to take any unsatisfiable formula requiring large rank in a very simple proof system, resolution, and derive new formulas that require large rank and tree-like proof size in these very powerful semi-algebraic systems. In particular, this construction applies to all formulas of large resolution width [4] since resolution width is a lower bound on resolution rank. A simplified statement of our main result is the following.

Theorem 1.1. *Let F be any family of polynomial-size 3-CNF formulas in m variables with resolution rank r. Then for any $\epsilon > 0$ and integer $1 \le k \le (1-\epsilon)\log\log n$, there is a family of polynomial-size CNF formulas $G = Lift_k(F)$ on $n = m^{O(k)}$ variables such that G requires $Th(k+1)$ refutations of rank $r^{\Omega(1)}/\log^{O(1)} n$ and tree-size $\exp(r^{\Omega(1)})$. In particular, if r is $m^{\Omega(1)}$ then G requires $Th(k+1)$ rank $n^{\Omega(1/k)}$, and tree-size $\exp(n^{\Omega(1/k)})$.*

Our lower bounds are much more general than this statement. In particular, our proof shows that the lifted formula requires large rank in any proof system in which the truth of each line in a proof can be verified by an efficient k-party randomized communication protocol; the above theorem follows by the reduction in [2]. Our lower bounds also apply to proof systems in which individual proof lines may not be efficiently verifiable but in which any falsifying assignment at an inferred line can be traced to one of its antecedents using an efficient k-party randomized communication protocol.

Our method is an example of a kind of hardness amplification that we term *hardness escalation*. The usual form of hardness amplification in circuit complexity is a method for amplifying the probability of error. That is, a complexity class C is fixed (or nearly fixed), and the goal is to go from a function that is weakly hard (e.g., any circuit in C that approximates f has non-negligible probability of error) to a function that is much harder (e.g., any circuit in a slightly smaller class than C that approximates f has error exponentially close to 1/2). With hardness escalation, we start with an object, a function f (or in this case an unsatisfiable 3-CNF formula), that is hard for some complexity class (a proof system here), and we construct a related function g (or lifted formula in this case) that is nearly as hard for a larger complexity class (respectively, more powerful proof system).

Our proof uses intuition and ideas from the pattern matrix method developed by Sherstov [35] and from a related method developed earlier by Raz and McKenzie [29]. Both of these are hardness escalation methods for communication complexity. Each method begins with a computational problem that is hard for a weak complexity measure, either a relation R of large decision tree complexity ([29]), or a function f of large approximate polynomial degree ([35]), and extends the problem using a "pattern matrix" to produce a problem of large deterministic ([29]), or large randomized and quantum ([35]), two-party communication complexity.

We use the k-party generalizations of the pattern matrix method developed in [24, 8]. Starting with a function f on m variables, and a parameter k, these generalizations *lift* f to obtain another function $g = \mathrm{Lift}_k(f)$ on mk bit-strings. The transformation replaces each original variable e_i by a Boolean *selector function* Ψ on k bit-strings. As long as f is hard in the weak measure, g is hard in the k-player number-on-forehead (NOF) randomized communication complexity model (for a particular partition of the new variables).

A key obstacle when trying to apply the pattern matrix method to the proof complexity setting is that the approach only works with Boolean functions, and not with unsatisfiable CNF formulas. To overcome this obstacle, we associate a family of Boolean functions $\mathcal{Z}_F$ with every unsatisfiable CNF F and show that if the hardness assumption on F is satisfied then there is some function $f \in \mathcal{Z}$ that has large decision tree complexity. Furthermore, if there is an efficient communication protocol that outputs falsified clauses in $\mathrm{Lift}(F)$, then there is an efficient protocol for $\mathrm{Lift}(f)$ for any $f \in \mathcal{Z}$. In this way we are able to combine the hardness escalation ideas of [29, 35] to obtain our results. We can also prove a converse to our result, thus characterizing the $\mathrm{Th}(k)$ rank of our lifted formulas. That is, in addition to deriving lower bounds on the rank of proofs for $\mathrm{Lift}_k(F)$ in terms of the resolution rank r of F, we also show that the rank of $T^{cc}(2)$ proofs (and even resolution) proofs of $\mathrm{Lift}_k(F)$ is not much larger than r.

Using the above lower bounds, we prove new rank separation theorems for hierarchies of polynomial threshold proof systems. By considering $\mathrm{Lift}_k(F)$ for certain unsatisfiable CNF formulas F that require large rank resolution refutations but need only small rank Cutting Planes refutations, we obtain strong rank separations between the power of $T^{cc}(k+1)$ and $R^{cc}(k)$, between $\mathrm{Th}(k)$ and $\mathrm{Th}(k-1)$, and between $\mathrm{CP}(k)$ and $\mathrm{CP}(k-1)$ refutations where $\mathrm{CP}(k)$ is the natural generalization of Cutting Planes to degree k.

Finally, using Sherstov's strengthened degree-discrepancy lemma for 2-player communication complexity [35], we apply our techniques to prove optimal integrality gaps for a large family of optimization problems even after n^{ϵ} rounds of CP or $\mathrm{Th}(1)$.

Due to space limitations some proofs are omitted or only sketched. Details appear in the full paper.

Related Work on Hardness Escalation. Hardness escalation results have been obtained in models such as communication complexity [35], sub-exponential time complexity [17], and circuit depth ([19, 12, 29]). A similar concept called hardness condensing was introduced in [6].

There have been a few papers in proof complexity implicitly using the idea of hardness escalation. First, several papers use the fact that if a formula F requires large resolution width, then the *xorification* of F, obtained by replacing each variable by an xor of several variables (and then rewriting as a CNF), is hard with respect to resolution size. [26] show how to replace variables in a somewhat hard unsatisfiable formula by hard functions in order to prove hardness escalation theorems for restricted tree-like proof systems. Lastly, Schoenebeck [30] obtains rank lower bounds for Lasserre proofs based on resolution rank lower bounds for particular families of formulas.

2. PRELIMINARIES

For a CNF formula F, let clauses(F) denote the set of clauses of F and let $|F|$ denote $|\text{clauses}(F)|$. F is a t-CNF formula iff every clause contains at most t literals. For any function f on m bits and any function h on s bits, we denote by $f \circ h$ the following function on ms bits: $f \circ h := f(h(\cdots), \cdots, h(\cdots))$.

Let F be an unsatisfiable CNF formula over variables $x_1, \ldots, x_n$ consisting of m clauses. The canonical Boolean relation associated with F is the predicate $R_F(x, y)$, where x is a vector of length n, and y is a number, $1 \le y \le m$. $R_F(\alpha, \beta)$ is true if and only if α is a Boolean assignment and the clause C_β in F is falsified by α. Associated with a Boolean relation $R(x, y)$ is a search problem: given x, output a y such that $R(x, y)$ is satisfied. Given a Boolean relation $R(x, y)$, we call a function g a *subfunction* of R if $R(x, g(x))$ is satisfied for every x. In other words, g is a particular function that solves the search problem associated with R. For example, for the canonical Boolean relation R_F associated with an unsatisfiable CNF formula F, the search problem, F_{search} is the problem of finding a violated clause given a Boolean assignment to the variables of F. A function g is a subfunction of R_F if for any truth assignment α, $g(\alpha)$ returns a clause of F that is falsified by α.

We adopt the usual notion of Boolean decision tree for a Boolean function. For a relation R, a decision tree T computes the search problem associated with R if it computes some subfunction of R. The *decision tree complexity* of f, denoted $D(f)$, is the minimum height of all decision trees computing f.

Hard Search Problems. Given an unsatisfiable CNF formula F that is somewhat hard (the decision tree complexity of F_{search}, $D(F_{\text{search}})$, is superpolylogarithmic in the size of F), we want to identify a set $\mathcal{Z}$ of Boolean functions associated with F that witnesses the hardness of F. Specifically, we want $\mathcal{Z}$ to have the property that if $D(F_{\text{search}})$ is large then $\mathcal{Z}$ contains a function with large decision tree complexity. This alone would be straightforward. However, we also want $\mathcal{Z}$ to be constructable from algorithms computing F_{search}.

A natural choice for the collection of functions from F_{search} would be to define $f_S(\alpha) = 1$ for some $S \subseteq \text{clauses}(F)$ if and only if there is some clause in S that is falsified by α. One might hope to argue that one such f_S would have decision tree complexity close to that of F_{search}. The obvious way to try to show this would be to reason by reduction; however, it is not clear how to construct a decision tree for F_{search} from decision trees for such a collection of f_S since both $f_S(\alpha)$ and $f_{\overline{S}}(\alpha)$ may equal 1. Some sort of symmetry-breaking scheme is required and this scheme must satisfy the property that for $S \subset T$ we have $f_T(\alpha) = 1$ whenever $f_S(\alpha) = 1$.

We say that a set $\mathcal{Z}$ of Boolean functions over the set of variables of an unsatisfiable CNF formula F is a *consistent system of functions for F* iff $\mathcal{Z} = \{f_S \mid S \subseteq \text{clauses}(F)\}$ and for any input assignment α there exists a clause C in F falsified by α such that for any $f_S \in \mathcal{Z}$ we have that $f_S(\alpha) = 1$ if and only if $C \in S$.

PROPOSITION 2.1. *Given an unsatisfiable CNF formula F, any function f^* that is a subfunction of R_F (that is, it solves the search problem F_{search}) yields a consistent system $\mathcal{Z}_{f^*}$ of functions for F.*

PROOF. Use the clause $C = f^*(\alpha)$ and define $f_S(\alpha) = 1$ iff $C \in S$. □

The following proposition says that any consistent system of functions for F witnesses the hardness of F.

PROPOSITION 2.2. *For any unsatisfiable CNF formula F and any consistent system $\mathcal{Z}$ of functions for F, there exists a function $f_S \in \mathcal{Z}$ such that $D(F_{\text{search}}) \le D(f_S)\lceil \log_2 |F| \rceil$.*

PROOF. Build a decision tree for F_{search} using binary search by querying the f_S for subsets $S \subseteq \text{clauses}(F)$ to narrow down the search. The requirement of consistency ensures that the path followed by binary search on input α yields the falsified clause C. To derive the tree for F_{search} replace each query of f_S by the optimal decision tree for f_S, yielding the claimed bound. □

Communication Complexity. Given a function (or relation) f, some number $k \ge 2$ of players, and a partition of the input of f for these players, we consider the *number-on-the-forehead (NOF)* communication model (cf.[22]), in which each player sees all inputs except the those in the block of the partition that is assigned to him, and the goal is compute f using a minimum of communication. Let $|\mathcal{P}|$ denote the maximum number of bits communicated in a communication protocol $\mathcal{P}$ and $\mathcal{P}(x)$ the output of the protocol on input x. A randomized protocol $\mathcal{P}$ is said to compute a function f with error at most ϵ if on any input x, with probability at least $1 - \epsilon$ (over the players' random coins c), $\mathcal{P}(x, c) = f(x)$.

If f is a search problem, the standard definition (e.g. [22]) of randomized communication complexity states that $\mathcal{P}$ computes f with error at most ϵ if and only if on any input x, for at least $1 - \epsilon$ fraction of the outcomes of random coins c, $\mathcal{P}(x, c) \in f(x)$. Among these good outcomes, the values $\mathcal{P}(x, c)$ may not be the same element in $f(x)$. However, for our construction, we require a stronger notion.

A randomized protocol $\mathcal{P}$ is said to *consistently compute* a relation f with error at most ϵ if there is a function f^* contained in f – that is, $f^*(x) \in f(x)$ for every x, such that $\mathcal{P}$ computes f^* with error at most ϵ.

Proof Systems and the Complexity of Clause Search. A wide variety of proof systems exist in the literature. In most of these proof systems, a proof or refutation can be expressed as a sequence of *lines*, each of which is either (a translation of) an input clause or follows from some previous lines via some sound *inference rule.* We call such proofs *standard proofs.* Similarly in a standard refutation system of an unsatisfiable formula f, a proof is again a asequence of lines, where the first line is f, the last line is the trivially false formula, and all other lines follow from some sound inference rule of the underlying proof system.

We associate a DAG $\mathcal{G} = (V, E)$ with every standard proof P, where V is the set of lines in P and $(u, v) \in E$ if line v is derived via some inference rule using line u. The *size* of P is the number of bits in P, which is lower-bounded by the number of lines in P. The *rank* of P is the length of the longest path in $\mathcal{G}$. We consider $\mathcal{G}$ to be a tree if every internal node has fanout one. (The axioms, which are not internal nodes, can be repeated.) If $\mathcal{G}$ is a tree, we say that P is *tree-like.* The *size complexity* and *rank complexity* of F in a standard proof system are the minimum size and minimum rank, respectively, of all proofs for F in that system. Similarly, we define *tree-like size complexity* as the minimum over all proofs are restricted to be tree-like.

Note that restricting a proof to be tree-like does not increase the rank of a proof because the same line can be rederived multiple times without affecting the rank. Tree-like size, however, can be much larger than general size.

In the most well-studied proof systems, there is a set of derivation rules (which can be thought of as inference schemas) of the form $F_1, F_2, \ldots, F_t \vdash G$ and each derivation step in a proof must be an instantiation of one of these rules. One such basic system is *resolution*, which manipulates clauses. Its only rule is the *resolution rule*: the clause $(A \vee B)$ is derived from $(A \vee x)$ and $(B \vee \neg x)$, where A and B are arbitrary disjunctions of literals and x is a variable. A resolution refutation of an unsatisfiable CNF formula f is a sequence of clauses, ending with the empty clause, such that each clause is either a clause of f, or follows from two previously derived clauses via the resolution rule. The well-known connection showing that DPLL executions and tree-like resolution proofs are equivalent gives us the following proposition.

PROPOSITION 2.3. [11] *For any CNF formula F, the minimum rank of any resolution proof of F is equal to $D(F_{\text{search}})$.*

Another proof system is the *Cutting Planes* (CP) proof system which manipulates integer linear inequalities. A CP refutation is a sequence of inequalities, ending with $0 \geq 1$, such that all other inequalities are either axioms ($0 \leq x_i$, $x_i \leq 1$), translated input clauses (input clause $(\ell_1 \vee \cdots \vee \ell_t)$ is translated as $\ell'_1 + \cdots + \ell'_t \geq 1$ where $\ell' = x$ if $\ell = x$ and $\ell' = (1 - x)$ if $\ell = \neg x$) or follow from two previously derived inequalities via either addition

$$p_1 \geq 0, \ldots, p_t \geq 0 \;\vdash\; \sum_{i=1}^{t} \lambda_i p_i \geq 0,$$

where each p_i is a linear form, or division with rounding

$$\sum_i c a_i x_i \geq b \;\vdash\; \sum_i a_i x_i \geq \lceil b/c \rceil,$$

where a_i, b, c, and $\lambda_i \geq 0$ are integers. There is a natural extension of CP, denoted CP(k), in which the above CP proof rules are extended to include p_i that are degree k multivariate polynomials and the x_i are replaced by degree k monomials. Since the input clauses are linear there are two other rules that allow the creation of higher degree inequalities, namely $p \geq 0 \;\vdash\; x_i p \geq 0$ and $p \geq 0 \;\vdash\; p \geq x_i p$ for all polynomials p of degree at most $k - 1$ and variables x_i. Other important well-studied proof systems are the Lovász-Schrijver proof systems (LS_0, LS, LS_+, and $\text{LS}_{+,\star}$) which manipulate polynomial inequalities of degree at most 2, and the Sherali-Adams and Lasserre proof systems which generalize the Lovász-Schrijver systems to higher degree.

Each of the above proof systems has a specific set of inference rule schemas, which allows them to have polynomial-time verifiers. We also consider more powerful *semantic proof systems* which restrict the form of the lines and the fan-in of the inferences but dispense with the requirement of a polynomial-time verifier and allow any semantically sound inference rule with a given fan-in. (Each line is a clause or follows via some semantic inference rule.) The fan-in must be restricted because the semantic rules are so strong.

For integer $k \geq 1$, we denote by Th(k) the semantic proof system, introduced in [2], whose proofs have fan-in 2, lines consist of polynomial inequalities of degree at most k, and input clauses and axioms are represented as linear inequalities as in the definition of CP above.

The following proposition follows from Caratheodory's Theorem. It is not hard to show that one can extend these simulations by Th(k) proofs to CP(k) and $LS^k_{+,\star}$.

PROPOSITION 2.4. *(1) Any CP proof of size (tree-like size) S and rank r can be converted to a* Th(1) *proof of size (tree-like size) $O(S)$ and rank $O(r \log n)$. (2) Any LS_0, LS, or* LS_+ *proof of size (tree-like size) S and rank r can be converted to a* Th(2) *proof of size (tree-like size) $O(S)$ and rank $O(r \log n)$.*

In this paper we also consider more general semantic proof systems than Th(k), namely those for which the fan-in is bounded and the truth value of each line can be computed by a multiparty communication protocol.

For any $k, C \geq 1$, we denote by $T^{cc}(k, C)$ the semantic proof system of fan-in 2 in which each proof line is a Boolean function whose value, for every partition of the input variables into k groups, can be computed by a C-bit randomized k-party NOF communication protocol of error at most 1/3. Both k and C may be integer functions of the input size of the formulas. In keeping with the usual notions of what constitutes efficient communication protocols, we use $T^{cc}(k)$ to denote the union of all $T^{cc}(k, C)$ over all C in $\log^{O(1)} n$.

Via standard boosting, we can replace the error 1/3 in the above definition by ϵ at the cost of increasing C by an $O(\log 1/\epsilon)$ factor. So, without loss of generality, in defining $T^{cc}(k)$ we can assume that the error is at most $1/n^{\log^{\Omega(1)} n}$.

Note also that a semantic proof of rank r that satisfies the same conditions as a $T^{cc}(k, C)$ proof except that it has rules of fan-in at most $t \geq 2$ can be simulated by a $T^{cc}(k, 2Ct \log_2 t)$ proof of rank $r \log_2 t$ by replacing each inference by a binary tree of height $\log_2 t$ in which lines of internal nodes are conjunctions of their predecessors.

For polylogarithmic k, the following lemma shows that Th(k) is a subclass of $T^{cc}(k + 1)$.

LEMMA 2.5. *For some constant $c > 0$, every* Th(k) *refutation of a CNF formula on n variables is also a $T^{cc}(k + 1, ck^3 \log^2 n)$ proof.*

PROOF. By the well-known result of Muroga [27], linear threshold functions on n Boolean values only require coefficients of $O(n \log n)$ bits. Since a degree k threshold polynomial is a linear function on at most n^k monomials, it is equivalent to a degree k threshold polynomial with coefficients of $O(kn^k \log n)$ bits. As shown in [2], over any input partition there is a randomized $(k+1)$-party communication protocol of cost $O(k \log^2 s)$ and error $\leq 1/s^{\Omega(1)}$ to verify a degree k polynomial inequality with s-bit coefficients. □

We also define another class of proofs based on k-party communication complexity that we will see is even more general than $T^{cc}(k, C)$. For any integer functions $k, C \geq 1$, we denote by $R^{cc}(k, C)$ the semantic proof system of arbitrary fan-in in which each proof line is a Boolean function such that the proof satisfies the following property: for every partition of the input variables into k groups, and every inference of B from $A_1, \ldots, A_s$ in the proof, there is a C-bit randomized k-party NOF communication protocol of error at most 1/3 that computes a (partial) function $f_{A_1,\ldots,A_s \vdash B}$ from the inputs to the set $[s]$ such that on every input α, if B evaluates to false on input α then $A_{f_{A_1,\ldots,A_s \vdash B}(\alpha)}$ evaluates to false on input α. We write $R^{cc}(k)$ to denote the union of all $R^{cc}(k, C)$ over all C in $\log^{O(1)} n$.

Lemma 2.6. *Every $T^{cc}(k, C)$ proof is an $R^{cc}(k, C)$ proof.*

Proof. The inferences in the $T^{cc}(k, C)$ are all of fan-in at most 2 and hence derive each line B from some lines A_1 and A_2. To compute the function $f_{A_1,A_2 \vdash B}$ the players evaluate A_1 on input α using the protocol given by the $T^{cc}(k, C)$ proof. If that evaluates to false then they output 1; otherwise, they output 2. □

We can sharpen this relationship further using a standard method for strengthening a proof system S by adding resolution rules over the lines of S [21]. Given a proof system S, we define related proof system $R(S)$ as follows: Lines of $R(S)$ are unordered disjunctions of lines of S and their negations. For every inference rule in S, $A_1, \ldots, A_t \vdash B$, there is the corresponding rule $(G \vee A_1), \ldots, (G \vee A_t) \vdash (G \vee B)$ where G is an arbitrary disjunction of lines of S and their negations. In addition there are extended resolution rules that allow the introduction of new disjuncts, $G \vdash (G \vee A_1 \vee \ldots \vee A_t)$, or cuts on lines of S, namely $(G \vee A), (H \vee \neg A) \vdash (G \vee H)$, where A is a line of S and G and H are arbitrary disjunctions of lines of S and their negations.

Lemma 2.7. *Every $R(T^{cc}(k, C))$ proof is an $R^{cc}(k, C)$ proof.*

Proof. For rules that correspond to rules of $T^{cc}(k, C)$ we apply the simple argument from Lemma 2.6 on the lines that are not common to all formulas. For the resolution rules, observe that the players only need to evaluate the line A to determine whether to select $(G \vee A)$ or $(H \vee \neg A)$. □

In particular, this shows via Lemma 2.5 that $R^{cc}(k + 1, ck^3 \log^2 n)$ proofs include the proof system $R(\text{Th}(k))$ (suggested by Hirsch). It is not clear whether one can efficiently simulate $R(\text{Th}(k))$ using $T^{cc}(k)$ proofs.

The following lemma, which is implicit in [2], gives the key relationships between $T^{cc}(k)$ and $R^{cc}(k)$ proofs and randomized communication protocols that consistently compute F_{search}.

Lemma 2.8. *Let $\epsilon > 0$ and F be an n-variable CNF formula.*

(i) If F has an $R^{cc}(k, C)$ refutation of rank r then, over any partition of the variables, there is an ϵ-error randomized k-party communication protocol $\mathcal{P}$ consistently computing F_{search} such that $|\mathcal{P}|$ is $O(Cr \log(r/\epsilon))$.

(ii) If F has a tree-like $T^{cc}(k, C)$ refutation of size S then, over any partition of the variables, there is an ϵ-error randomized k-party protocol $\mathcal{P}$ consistently computing F_{search} such that $|\mathcal{P}|$ is $O(C \log S \log \frac{\log S}{\epsilon})$.

Proof. First assume that we have a rank r refutation in $R^{cc}(k, C)$. On input α, the k players backtrack from the last derived inequality in the proof $(0 \geq 1)$ to find some clause that is falsified by α. When they are at a line B that follows from lines $A_1, \ldots, A_s$ in the proof, they run the protocol for $f_{A_1,\ldots,A_s \vdash B}$, implied by the $R^{cc}(k, C)$ definition for the inference at B, $O(\log(r/\epsilon))$ times and take the majority answer to reduce its error below ϵ/r. Then the players move to the line indicated by that answer. The probability that this protocol makes an error is at most the sum of all error probabilities on any path in the proof. Since the last line evaluates to false on input α, in the case that there is no error the players will return a fixed clause in the proof that is falsified by α, which implies that they consistently compute F_{search}.

For the second case of a size S tree-like refutation, there is some line in the refutation that is derived from between $S/3$ and $2S/3$ of the lines of the refutation tree. The players first evaluate that line with error at most $\epsilon/(2 \log_2 S)$ by repeating the protocol $O(\log(\log S/\epsilon))$ times. If the line evaluates to false then they continue within that subtree; otherwise, they remove the nodes of that subtree. This is done recursively until a falsified clause is found. The depth of recursion is at most $2 \log_2 S$. The rest is similar to the first case. □

3. HARDNESS ESCALATION

The high level idea of our method of hardness escalation is to take a somewhat hard an unsatisfiable t-CNF formula F (one for which F_{search} requires a large height decision tree) over variables $e_1, \ldots, e_m$, and build a new CNF formula $G = \text{Lift}(F)$ of size $m^{O(t)}$ by lifting F using some function ψ that encodes e_i using a larger collection of input bits. This lifting over CNF is adapted from previous work for Boolean functions, which we review next.

3.1 Lifting Decision Tree Complexity

Given $k, s > 0$ and a domain A, a function $\psi_k : \{0,1\}^s \times A^k \mapsto \{0,1\}$ is called a *selector* if there is some $h : A^k \mapsto [s]$ such that $\psi_k(x, y_1, \ldots, y_k) = x_{h(y_1,\ldots,y_k)}$ for every $x \in \{0,1\}^s$ and $y_i \in A$. Informally, ψ_k outputs a bit in x that is selected by the values of $y_1, \ldots, y_k$.

We consider two specific selector encodings ψ_k: the tensor selector $\psi^{\text{T}}_{k,\ell}$ and the parity selector $\psi^{\oplus}_{k,a}$. For the tensor selector $\psi^{\text{T}}_{k,\ell}(x, y)$, we have $s = \ell^k$ and $A = [\ell]$, and we think of $x \in \{0,1\}^s$ as indexed by A^k and hence $h(\cdot)$ is just the identity function on A^k. For the parity selector $\psi^{\oplus}_{k,a}(x, y)$, we have $s = 2^a$, $A = \{0,1\}^a$, and we think of x as indexed by a-bit arrays and $h(y_1, \ldots, y_k) = y_1 \oplus \cdots \oplus y_k$.

Given an initial function f over variables x, define g, the $(k+1)$-lifted version of f, to be the function $f \circ \psi_k$.

It is not hard to see that if the decision tree complexity of f is d, then for any $k \geq 2$, and over any partition of the variables into k groups, there is a k-party communication protocol computing g of cost approximately $d \cdot c$, where c is the cost of computing ψ_k. The k players just simulate the decision tree for f and the cost of computing any single variable in f encoded by ψ_k is c bits. If ψ_k is simple enough, and therefore c is negligible, then this cost is approximately equal to d. Ideally, we would like to argue that this is the best that the players can do. Intuitively, since we have encoded each input bit in f indirectly, the players need to communicate $\Omega(1)$ bits in order to be able to "learn" any single bit. If the decision tree complexity of f is large, we would hope that g has large communication complexity. Recent results in communication complexity show that we cannot do much better than the above trivial protocol, subject to some constraints on ψ_k.

We need the following approximation notion to bridge decision tree complexity and communication complexity. Given any $0 \leq \epsilon < 1$, the *ϵ-degree* of a real-valued function f, $\deg_\epsilon(f)$, is the smallest d for which there exists a multivariate real-valued polynomial p of degree d such that $||f - p||_\infty = \max_x |f(x) - p(x)| \leq \epsilon$. This notion of approximating a real-valued function is polynomially related to decision tree complexity.

Proposition 3.1. [28, 1] *For every Boolean function f,* $\deg_{5/6}(f) \leq D(f) \leq (4 \deg_{5/6}(f))^6$.

Finally we state the communication lower bounds for $g = f \circ \psi_k$. The following input partition is always assumed when the communication complexity of g is discussed: there are $k+1$ players and for each input $(x, y_1, \ldots, y_k)$ to each ψ_k, player 0 is assigned x, and each player i, for $1 \leq i \leq k$, is assigned y_i. Intuitively, the inputs $y_1, \ldots, y_k$ given to players 1 through k determine which bits of x are given to f. The next two results say that, when ψ_k is either $\psi^{\mathrm{T}}_{k,\ell}$ or $\psi^{\oplus}_{k,a}$, and the encoding ψ_k is over a large enough number of new variables, then the communication complexity of g is polynomial related to $D(f)$ (up to a factor depending only on k).

Theorem 3.2. [8] *Let $f : \{0,1\}^m \mapsto \{0,1\}$ with 5/6-degree $d > 2$. If $\ell > \frac{2^{2^{k+1}} kem}{d}$ then any $(k+1)$-party communication protocol $\mathcal{P}$ computing $g = f \circ \psi^{\mathrm{T}}_{k,\ell}$ with error 1/3 must have $|\mathcal{P}| = \Omega(\frac{d}{2^k})$.*

Theorem 3.3. [3] *Let $f : \{0,1\}^m \mapsto \{0,1\}$ with 5/6-degree $d > 2$. If $2^a > \frac{2^{2^{k+1}+2k} em}{d}$ then any $(k+1)$-party communication protocol $\mathcal{P}$ computing $g = f \circ \psi^{\oplus}_{k,a}$ with error 1/3 must have $|\mathcal{P}| = \Omega(\frac{d}{2^k})$.*

The first theorem uses the tensor selector while the second uses the parity selector. We will use both to prove lower bounds for $T^{cc}(k)$ and $R^{cc}(k)$ proof systems. The parity selector requires fewer bits to encode each variable, thus leading to stronger proof complexity lower bounds as a function of the number of variables (though it is no more efficient with respect to formula size). In contrast, the lifted CNF formula derived using the tensor selector is easier to refute by small degree threshold proofs, allowing us to prove rank separations for the hierarchies of $T^{cc}(k)$ and $R^{cc}(k)$ proof systems.

Overview of the Hardness Escalation Argument. Before giving the formal construction and proofs for the two selectors, we present a brief overview of our argument. Let F be any t-CNF over the variables $e_1, \ldots, e_m$. To lift F to obtain a harder unsatisfiable formula G, every variable e_i of F will be replaced by a set of variables V_i. The V_i variables be comprised of $k+1$ sets of variables: x, and $y_1, \ldots, y_k$. A selector function ψ_k will use the y variables to select one x variable to represent e_i. The clauses in G will state that the V_i variables represent a valid ψ-encoding, and that with respect to this encoding, F is true. By Lemma 2.8, we know that if G has low $T^{cc}(k)$ rank, then there is an efficient $(k+1)$-party protocol for solving the search problem associated with G, G_{search}. Thus to prove a $T^{cc}(k)$ rank lower bound for G, it suffices to prove that G_{search} is hard in the $(k+1)$-party NOF model.

Now any function associated with G is also a lifting of the corresponding function associated with F. In particular, $G_{\text{search}} = F_{\text{search}} \circ \psi_k$. The intuition for why it should be hard is similar to that of the lifting of Boolean functions: here G_{search} is a lifting of F_{search}, and the decision tree complexity of F_{search} is large. To prove this, assume for sake of contradiction that G_{search} is easy for $(k+1)$-party communication. Then the players can efficiently compute G_{search} over the variables V_i. This in turn means that given the variables V_i, they can efficiently compute $F_{\text{search}}(e_1, \ldots, e_m)$, where each $e_i = \psi_k(V_i)$. It follows that there exists a consistent system $\mathcal{Z}$ of functions for F such that for any function $f_S \in \mathcal{Z}$, the players can easily compute $f_S \circ \psi_k$. In other words, the lifting of any $f_S \in \mathcal{Z}$ is easy for k-party communication. It then follows that for appropriate choices of ψ_k, any function in $\mathcal{Z}$ has low decision tree complexity. Then by Proposition 2.2, we can conclude that the decision tree complexity of F_{search} is small, contradicting our assumption. We now proceed to the formal arguments for each of the selector functions.

3.2 Hardness using the Tensor Selector

Let F be any t-CNF over the variables $e_1, \ldots, e_m$. Given $k, \ell \geq 2$, $G = \text{Lift}^{\mathrm{T}}_{k,\ell}(F)$ is a CNF formula defined over m sets of variables $V_1, \ldots, V_m$, where each V_i is further partitioned into two sets X_i of size ℓ^k and Y_i of size $k\ell$. Intuitively, every V_i is an encoding of e_i based on $\psi^{\mathrm{T}}_{k,\ell}$. Each X_i represents a k-dimensional tensor of size ℓ^k each of whose cells c is associated with a variable $x_{i,c} \in X_i$. Y_i, which is indexed as $\{y_{i,p,a} : 1 \leq p \leq k, 1 \leq a \leq \ell\}$, selects a unique cell c in this tensor as follows: For each $p \in [k]$, exactly one of the variables $y_{i,p,a}$ for $a \in [\ell]$ is true, and the value a_p such that y_{i,p,a_p} is true is the p-th coordinate of c. Every clause in F is then transformed into a set of clauses over these V_i. Formally, the clauses in G consist of:

- For $1 \leq i \leq m, 1 \leq j \leq k$, exactly one of $y_{i,p,1}, \ldots, y_{i,p,\ell}$ is 1:
  (I) $y_{i,p,1} \vee \cdots \vee y_{i,p,\ell}$
  (II) $(1 \leq a < a' \leq \ell)$: $\neg y_{i,p,a} \vee \neg y_{i,p,a'}$
- For every clause, say $\neg e_{i_1} \vee e_{i,2} \vee \cdots \vee e_{i_t}$, in F and for every t-tuple of cells $(c_1, \ldots, c_t)$, if Y_{i_1} selects c_1, Y_{i_2} selects c_2, etc., then $\neg x_{i_1,c_1} \vee x_{i_2,c_2} \cdots \vee x_{i_t,c_t}$ must be satisfied: this is translated into one clause of $tk + t$ literals. For example, if the coordinates of $c_1, \ldots, c_t$ are $(a^1_1, \ldots, a^1_k), \ldots, (a^t_1, \ldots, a^t_k)$, respectively, then the clause would be:
  (III) $\neg y_{i_1,1,a^1_1} \vee \cdots \vee \neg y_{i_1,k,a^1_k} \vee \cdots \vee \neg y_{i_t,1,a^t_1} \vee \cdots$
  $\vee \neg y_{i_t,k,a^t_k} \vee \neg x_{i_1,c_1} \vee x_{i_2,c_2} \vee \cdots \vee x_{i_t,c_t}$

The next proposition shows that as long as the clauses of F are not too large, then G is also not too large, and that G is unsatisfiable as long as F is.

Proposition 3.4. *If F is a t-CNF over m variables, then $G = \mathrm{Lift}^{\mathrm{T}}_{k,\ell}(F)$ is a CNF formula of $|F|\ell^{tk}+O(mk\ell^2)$ clauses of size at most $\max\{tk+t,\ell\}$ over $n = m(\ell^k+k\ell)$ variables. Furthermore, if F is unsatisfiable, then so is G.*

We say that an assignment to $X_1, Y_1, \ldots, X_m, Y_m$ of G is a *valid encoding* of an assignment to variables $e_1, \ldots, e_m$ of F if all clauses (I) and (II) are satisfied and for every i, $x_{i,c} = e_i$ where c is selected by Y_i.

We fix the following input partition to $k+1$ players when discussing the communication complexity of G_{search}: player 0 is assigned all of the X_i's, and each player p, for $1 \le p \le k$, is assigned $\{y_{i,p,a} : 1 \le i \le m, 1 \le a \le \ell\}$.

The following lemma says that if G_{search} is easy in communication complexity, then there exists a consistent system $\mathcal{Z} = \{f_S : S \subseteq \mathrm{clauses}(F)\}$ for F such that for every $f_S \in \mathcal{Z}$, computing $f_S \circ \psi^{\mathrm{T}}_{k,\ell}$ is also easy in communication complexity.

Lemma 3.5. *Given any unsatisfiable t-CNF formula F and $G = \mathrm{Lift}^{\mathrm{T}}_{k,\ell}(F)$. Suppose that there is a $(k+1)$-party communication protocol $\mathcal{P}$ consistently computing G_{search} with error ϵ such that $|\mathcal{P}| \le C$. Then there exists a consistent system $\mathcal{Z} = \{f_S : S \subseteq \mathrm{clauses}(F)\}$ of functions for F such that for every S, there is a $(k+1)$-party communication protocol $\mathcal{P}_S$ consistently computing $f_S \circ \psi^{\mathrm{T}}_{k,\ell}$ with error ϵ such that $|\mathcal{P}_S| \le C$.*

Proof. For every input assignment α to F, we fix any input assignment α^{T} to G that is a valid encoding of α. Let g^* be the subfunction of G_{search} that is computed by $\mathcal{P}$.

We first observe that on any input assignment α and α^{T}, $g^*(\alpha^{\mathrm{T}})$ always outputs a type (III)-clause. This is because α^{T} is a valid encoding. This clause corresponds to a unique clause in F that is falsified by α. Thus g^* uniquely determines a subfunction f^* of F_{search}.

Given f^*, we define the consistent system $\mathcal{Z} = \mathcal{Z}_{f^*}$ for F using the construction in Proposition 2.1. For every $f_S \in \mathcal{Z}$, the protocol $\mathcal{P}_S$ for $f_S \circ \psi^{\mathrm{T}}_{k,\ell}$ is adapted from $\mathcal{P}$ in the straightforward way. □

The next theorem is our main result on the proof complexity of G which glues all the parts together.

Theorem 3.6. *There are absolute constants $c, c' > 0$ such that the following holds. Let F be any t-CNF formula on m variables having resolution rank at least r and let $G = \mathrm{Lift}^{\mathrm{T}}_{k,\ell}(F)$ for $\ell \ge \frac{c2^{2^{k+1}}km}{(r/\log|F|)^{1/6}}$. Then for any C and $M = c'(r/\log_2|F|)^{1/6}/(C2^k)$,*

- *any $R^{cc}(k+1, C)$ refutation of G of rank R must have $R\log_2 R \ge M$, and*
- *any tree-like $T^{cc}(k+1, C)$ refutation of G of size S must have $\log S \log\log S \ge M$.*

Proof. We will prove only the first part; the second part follows similarly. Let P be a $R^{cc}(k+1, C)$ refutation of G of rank R. Lemma 2.8, there exists a $(k+1)$-party protocol $\mathcal{P}$ consistently computing G_{search} of error $1/3$ such that $|\mathcal{P}|$ is $O(CR\log R)$.

Now on the one hand, by Lemma 3.5, there exists a consistent system $\mathcal{Z} = \{f_S : S \subseteq \mathrm{clauses}(F)\}$ of functions for F such that for every S, there exists a $(k+1)$-party protocol $\mathcal{P}_S$ computing $f_S \circ \psi^{\mathrm{T}}_{k,\ell}$ of error $1/3$ such that $|\mathcal{P}_S|$ is $O(CR\log R)$.

On the other hand, by Proposition 2.3, the assumption on the resolution rank of F implies that $D(F_{\mathrm{search}}) \ge r$. By Proposition 2.2, there exists a function $f_S \in \mathcal{Z}$ such that

$$D(f_S) \ge \frac{D(F_{\mathrm{search}})}{\lceil \log_2|F| \rceil} = \frac{r}{\lceil \log_2|F| \rceil}.$$

By Proposition 3.1, we have $d = \deg_{5/6}(f_S) \ge (D(f_S))^{1/6}/4 \ge (\frac{r}{\log_2|F|})^{1/6}/4$.

Finally, by Theorem 3.2, we must have $CR\log R$ that is $\Omega(d/2^k)$ which is $\Omega((r/\log_2|F|)^{1/6}/2^k)$. □

We note that we have a somewhat matching upper bound on the rank complexity of G.

Lemma 3.7. *Let F be a t-CNF formula on m variables having resolution rank r. Then for any $\ell \ge 1$, there is an $T^{cc}(2, \log_2(rk\ell))$ proof of $G = \mathrm{Lift}^{\mathrm{T}}_{k,\ell}(F)$ of rank at most $crk\log_2\ell$, where $c > 0$ is some absolute constant.*

Proof Sketch. The main idea is to first build a decision tree (over linear inequalities) for G_{search} using the decision tree for F_{search}, and then convert this decision tree to a $T^{cc}(2)$ refutation. For every e_i there is precisely one variable $x_{i,(a_1,\ldots,a_k)}$ whose value will replace that of e_i in evaluating G. This selection is determined by the one tuple for which all of $y_{i,1,a_1}, \ldots, y_{i,k,a_k}$ evaluate to 1. Whenever the decision tree for F_{search} queries a variable e_i, the decision tree for G_{search} does k binary searches to find this tuple. Then the query of e_i is replaced by a query to $x_{i,(a_1,\ldots,a_k)}$.

The second step is carried out by following the standard conversion of decision trees to proofs implicit in the equivalence in Proposition 2.3. The conversion produces a binary proof tree where each line can be viewed as a disjunction of two linear inequalities on at most $rk\ell$ variables which can be evaluated efficiently by a 2-party randomized protocol. □

The following corollary, which implies Theorem 1.1, follows easily from Theorem 3.6, Lemma 3.7, and Proposition 3.4.

Corollary 3.8. *Let t be some constant. Suppose that a family of polynomial-size t-CNF formulas F on m variables has resolution rank complexity $r = r(m)$. Then, for every constant $\epsilon > 0$ and $k \le (1-\epsilon)\log\log n$, there is a family of CNF formulas $G = \mathrm{Lift}^{\mathrm{T}}_k(F)$ on $n = m^{O(k)}$ variables of size $n^{O(t)}$ such that*

- *G requires $R^{cc}(k+1)$ rank complexity $\Omega(r^{1/7})$;*
- *there is a $T^{cc}(2)$ refutation of G of rank $O(r\log n)$;*
- *G requires $T^{cc}(k+1)$ tree-size $\exp(\Omega(r^{1/7}))$.*

3.3 Hardness using the Parity Selector

Let F be any t-CNF over the variables $e_1, \ldots, e_m$. Given $k, a \ge 2$, $\mathrm{Lift}^{\oplus}_{k,a}(F)$ is a CNF defined over m sets of variables $V_1, \ldots, V_m$, where each V_i is further partitioned into two sets X_i and Y_i. The difference here with $\mathrm{Lift}^{\oplus}_{k,a}(F)$ is that every V_i is an encoding of e_i based on $\psi^{\oplus}_{k,a}$. That is, each X_i has 2^a variables that are indexed by a-bit vectors, each $Y_i = \{y_{i,p,b} : 1 \le p \le k, 1 \le b \le a\}$ has ka variables, and each Y_i selects a unique a-bit vector c with $c_b = \oplus_{p=1}^k y_{i,p,b}$, for $1 \le b \le a$. The clauses in $\mathrm{Lift}^{\oplus}_{k,a}(F)$ consist of:

(*) For every clause, say $e_{i_1} \vee \cdots \vee e_{i_t}$, in F and for every t-tuple of a-bit vectors $(c_1, \ldots, c_t)$, if Y_{i_1} selects c_1, Y_{i_2} selects c_2, etc., then $x_{i_1,c_1} \vee \cdots \vee x_{i_t,c_t}$ must be satisfied. For every clause and t-tuple, this is translated into $\leq 2^{tka}$ clauses of size $tka + t$ in the straightforward way. That is, there are $\leq 2^{ka}$ assignments to the bits in Y_{i_1} that make them select c_1, and similarly for Y_{i_2}, etc. There is one clause, similar to the clauses of type (III) in the tensor selector case, corresponding to each such assignment.

Proposition 3.9. *If F is a t-CNF over m variables, then $G = \mathrm{Lift}^{\oplus}_{k,a}(F)$ is a CNF formula of at most $|F|2^{tka+ta}$ clauses of size at most $ka + t$ over $n = m(2^a + ka)$ variables. Furthermore, if F is unsatisfiable, then so is G.*

The rest of the proofs for this section are very similar to those in the last section. The first difference is that since $\psi^{\oplus}_{k,a}$ gives a more efficient encoding than $\psi^{\mathrm{T}}_{k,\ell}$, the blow-up in the number of variables of G is significantly reduced. The second difference is that, here, G has a small rank resolution refutation, as opposed to a $CC(2)$ refutation in the last section when the lifting was done using tensor-encoding. There, small rank resolution refutation was impossible because the final clauses were too large.

By using Theorem 3.3 for $\psi^{\oplus}_{k,a}$ in place of Theorem 3.2 we can replace the function $G = \mathrm{Lift}^{\mathrm{T}}_{k,\ell}(F)$ in Theorem 3.6 by $G = \mathrm{Lift}^{\oplus}_{k,a}(F)$ and obtain exactly the same conclusion provided that $2^a > (c2^{2^{k+1}+2k}m)/(r/\log|F|)^{1/6}$.

On the other hand, one can upper bound the rank complexity of $\mathrm{Lift}^{\oplus}_{k,a}(F)$ in terms of that of F, even in resolution.

Lemma 3.10. *Let F be a t-CNF formula on m variables having resolution rank r. There is some absolute constant $c > 0$ such that for any $a \geq 1$, there is a resolution refutation of $G = \mathrm{Lift}^{\oplus}_{k,a}(F)$ of rank at most $crka$.*

Corollary 3.11. *Let t be some constant. Suppose that a family of polynomial size t-CNF formulas F on m variables has resolution rank complexity $r = r(m)$. Then, for every $\epsilon > 0$ and $k \leq (1-\epsilon)\log\log n$, there is a family of CNF formulas $G = \mathrm{Lift}^{\oplus}_{k}(F)$ on $n = m^{O(1)}$ variables of size $n^{O(tk)}$ such that*

- *G has $R^{cc}(k+1)$ rank complexity $\Omega(r^{1/7})$;*
- *there is a resolution refutation of G of rank $O(rk\log n)$;*
- *G requires $T^{cc}(k+1)$ tree-size $\exp(\Omega(r^{1/7}))$.*

4. PROOF SYSTEM HIERARCHIES

In this section we separate $R^{cc}(k)$ and $\mathrm{CP}(k)$ in terms of rank and separate $T^{cc}(k)$ and $\mathrm{CP}(k)$ in terms of tree-like size, thereby separating $R^{cc}(k+1)$ from $R^{cc}(k)$ and $T^{cc}(k+1)$ from $T^{cc}(k)$. In particular we show that if an unsatisfiable t-CNF formula F has a small rank CP proof, then $G = \mathrm{Lift}^{\mathrm{T}}_{k-1,\ell}(F)$ has a small rank $\mathrm{CP}(k)$ proof (that can be made small and tree-like). Moreover, if F requires large resolution rank, then with the right parameters, G has no small rank $R^{cc}(k)$ (or small tree-like $T^{cc}(k)$) proof. Thus G is a separating instance.

The pigeonhole principle is known to be hard for resolution but admits a small rank CP proof. Since we need the clauses of the input formula to be of constant size for the size of the formula $\mathrm{Lift}^{\mathrm{T}}_{k,\ell}(F)$ to be polynomial, we use the following generalization of the pigeonhole principle [4].

Let $\mathcal{G} = (U \cup V, E)$ be any bipartite graph, where U represents the pigeons and V the holes and associate a variable $0 \leq e_{(u,v)} \leq 1$ with each edge $(u,v) \in E$. $\mathcal{G}-\mathrm{PHP}$ consists of the following clauses, which have been translated to inequalities:

(P) for all $u \in U : \sum_{(u,v)\in E}\ e_{(u,v)} \geq 1$

(H) for all $u \neq u' \in U, v \in V$ s.t. $(u,v), (u',v) \in E$: $e_{(u,v)} + e_{(u',v)} \leq 1$

Proposition 4.1. [4] *For every n, there is a bipartite graph $\mathcal{G} = (U \cup V, E)$, where $|U| = |V| + 1 = n$ and the degree of every vertex in U is ≤ 5, such that $\mathcal{G}-\mathrm{PHP}$ is a polynomial size 5-CNF on $m = 5n$ variables and requires resolution rank $\Omega(m)$.*

From this and Corollary 3.8, we immediately obtain a rank lower bound for a lifting of $\mathcal{G}-\mathrm{PHP}$.

Lemma 4.2. *There is a family of bipartite graphs $\mathcal{G}$ and a family of polynomial-size CNF formulas $\mathrm{Lift}^{\mathrm{T}}_{k-1}(\mathcal{G}-\mathrm{PHP})$ on n variables that requires $R^{cc}(k)$ rank $n^{\Omega(1/k)}$ and $T^{cc}(k)$ tree-like refutation size $\exp(n^{\Omega(1/k)})$ for $k \leq (1-\epsilon)\log\log n$ and any constant $\epsilon > 0$.*

Our upper bound for the lifted versions of $\mathcal{G}-\mathrm{PHP}$ will be derived from the following CP rank upper bound for $\mathcal{G}-\mathrm{PHP}$ itself.

Proposition 4.3. [5] *For any $\mathcal{G} = (U \cup V, E)$ with $|U| = |V| + 1$, $\mathcal{G}-\mathrm{PHP}$ has a CP refutation of rank $O(\log|U|)$.*

Before considering the lifted version of $\mathcal{G}-\mathrm{PHP}$, we define some convenient $\mathrm{CP}(k)$ consequences for lifted formulas. Suppose that F has variables $e_1, \ldots, e_m$ and let $G = \mathrm{Lift}^{\mathrm{T}}_{k-1}(F)$. The variables in G are $x_{i,c}$ (each cell c is indexed by a tuple in $[\ell]^{k-1}$) and $y_{i,p,a}$, where $1 \leq i \leq m$, $1 \leq p \leq k-1$, and $1 \leq a \leq \ell$. For each variable e_i of F define a degree k polynomial

$$\mathbf{e_i} := \sum_{c=(a_1,\ldots,a_k)\in[\ell]^{k-1}} x_{i,c}\mathbf{y_{i,c}},$$

where $\mathbf{y_{i,c}} := y_{i,1,a_1} \cdot y_{i,2,a_2} \cdots y_{i,k-1,a_{k-1}}$ and then define the following forms:

(I') for all $1 \leq i \leq m$: $\sum_{c\in[\ell]^{k-1}} \mathbf{y_{i,c}} \geq 1$

(II') for all $1 \leq i \leq m$ and $c \neq c' \in [\ell]^{k-1}$: $\mathbf{y_{i,c}} + \mathbf{y_{i,c'}} \leq 1$

(III') for all clauses in F, say $\neg e_{i_1} \vee e_{i_2} \vee \cdots \vee e_{i_t}$, and for every t-tuple of cells $(c^1, \ldots, c^t)$,

$$\mathbf{y_{i_1,c^1}} x_{i_1,c^1} + \mathbf{y_{i_2,c^2}}(1 - x_{i_2,c^2}) + \cdots + \mathbf{y_{i_t,c^t}}(1 - x_{i_t,c^t}) \leq t - 1$$

Lemma 4.4. *For any $k, \ell \geq 2$ and any CNF formulas F and $G = \mathrm{Lift}^{\mathrm{T}}_{k-1,\ell}(F)$, given the families of clauses (I), (II), and (III) in G, there are $\mathrm{CP}(k)$ derivations of rank k of all (I'), (II'), and (III') inequalities as well as $0 \leq \mathbf{y_{i,c}} \leq 1$ and $0 \leq \mathbf{e_i} \leq 1$.*

Lemma 4.5. *For any $\mathcal{G} = (U \cup V, E)$ with $|U| = |V| + 1$ and the degree of every vertex in U is at most t, the formula $G = \mathrm{Lift}^{\mathrm{T}}_{k-1,\ell}(\mathcal{G}-\mathrm{PHP})$ has a $\mathrm{CP}(k)$ refutation of rank $O(\log|U| + tk\log\ell)$, for any $k, \ell \geq 2$.*

Proof Sketch. For ease of notation, we denote the variables in $F = \mathcal{G}-\mathrm{PHP}$ by $e_1, \ldots, e_m$ where $m = |E|$. We apply Lemma 4.4 to obtain a rank k derivation of the inequalities (I'), (II'), (III') and $0 \leq \mathbf{e_i} \leq 1$ from the original inequalities for G. Given the inequalities (I'),(II'), and (III') we then show that for each (P)-type axiom $e_{i_1} + \ldots + e_{i_t} \geq 1$ (for some $t > 0$) in F, we can derive the corresponding inequality $\mathbf{e_{i_1}} + \ldots + \mathbf{e_{i_t}} \geq 1$ via a rank-$O(tk \log \ell)$ proof in $\mathrm{CP}(k)$. Given inequalities (I'),(II'), and (III'), we also show that for all (H) axioms $e_{i_1} + e_{i_2} \leq 1$ in F, $\mathbf{e_{i_1}} + \mathbf{e_{i_2}} \leq 1$ is derivable via a rank-$O(k \log \ell)$ proof in $\mathrm{CP}(k)$. Finally, we simulate the CP-refutation for F in Proposition 4.3 with each variable e_i replaced by the degree k polynomial $\mathbf{e_i}$. □

We remark that the construction in Lemma 4.5 can be similarly applied to any unsatisfiable t-CNF formula F to show that if F has small rank CP proof, then $\mathrm{Lift}^{\mathrm{T}}_{k-1,\ell}(F)$ has a small rank $\mathrm{CP}(k)$ proof. Lemmas 4.2 and 4.5 together yield the following separations of our proof systems.

Theorem 4.6. *For any $\epsilon > 0$ there is a family of unsatisfiable CNF formulas G on n variables that requires nearly polynomial rank $n^{\Omega(1/\log\log n)}$ in all $R^{cc}(k)$ systems and nearly exponential tree-like size $\exp(n^{\Omega(1/\log\log n)})$ in all $T^{cc}(k)$ systems, but has logarithmic rank and polynomial tree-like size in $\mathrm{CP}(k)$ systems, for any $k \leq (1-\epsilon)\log\log n$.*

5. INTEGRALITY GAPS

The MAX-SAT problem is well-studied in the theory of approximation algorithms and optimal inapproximability results are known under the assumption that $\mathsf{P} \neq \mathsf{NP}$. There are also unconditional inapproximability results known for a restricted class of algorithms that involve applying Cutting Planes or LS+ procedures to a relaxation of the standard integer program (e.g. [5, 30]). Here we use our lifting approach to derive optimal unconditional inapproximability results for small rank Th(1) proofs.

Given a CNF formula $G = \{C_1 \wedge \cdots \wedge C_m\}$ over variables $x_1, \ldots, x_n$, we can add a new set of variables $z_1, \ldots, z_m$, and define $C'_i = \neg z_i \vee C_i$. Let G' be $C'_1 \wedge \cdots \wedge C'_m$. If we convert these clauses into linear constraints and add Boolean constraints, we obtain a linear program L_G with objective function $\sum_i z_i$ that is a natural LP relation of the MAX-SAT problem for G.

There are $2^t\binom{n}{t}$ clauses over n variables that contain exactly t different variables. Let $\mathcal{N}^{t,n}_m$ be the probability distribution induced by choosing m of these clauses uniformly and independently.

Let F be a t-CNF formula. We consider $G = \mathrm{Lift}^{\mathrm{T}}_{1,2}(F)$ as described earlier, except that since we have set $k = 1$ and $\ell = 2$, the form of G can be considerably simplified. That is, the variables of G will consist of two bit-vectors, x and y, in which x will contain n blocks, each of size 2, y will be a vector of length n, where y_i indicates which of the two elements of block i will be chosen in x. Each clause of F is transformed into 2^t clauses in G, corresponding to each of the 2^t possible bits of x that could be chosen by y. Thus if F has m clauses, each of size t, G has $2^t m$ clauses, each of size $2t$, and F is unsatisfiable if and only if G is unsatisfiable. (There is no need for the clauses on the y_i-variables that were used in the case of larger ℓ.)

The following theorem, which is key to getting an integrality gap, is a quantitatively stronger version of Theorem 3.2, for the case $k = 1$ and $\ell = 2$.

Theorem 5.1. *Let F be any t-CNF formula on n variables having resolution rank at least r, and let $G = \mathrm{Lift}^{\mathrm{T}}_{1,2}(F)$. Then any $R^{cc}(2, C)$ refutation of G of rank R must have $CR \log R \geq r^{\delta}$ for some constant $\delta > 0$.*

The proof of Theorem 5.1 relies on the following stronger version of Theorem 3.2 for the special case of 2 players due to Sherstov [35]. Theorem 3.2 requires that ℓ be large. The theorem below has much less dependence on the degree, and as a result it does not require ℓ to be large. However, this quantitatively stronger version is currently only known to hold for 2-player communication complexity.

Theorem 5.2. [35] *Let f be a boolean function on n variables with sign-degree at least d (and hence $5/6$-degree at least d). Then any 2-party communication protocol $\mathcal{P}$ computing $g = f \circ \psi^{\mathrm{T}}_{1,2}$ with error 1/3 must have $|\mathcal{P}| = \Omega(d)$.*

The proof of Theorem 5.1 is similar to that of Theorem 3.6, using Theorem 5.2 instead of Theorem 3.2, and is omitted.

We now see how the above theorem can be applied to derive an integrality gap for small rank Th(1) or Cutting Planes proofs.

Corollary 5.3. *Let $t \geq 3$ be an integer. There exists $\delta < 1$ such that for all $\epsilon > 0$ there is a $\Delta > 1$ such that for a randomly chosen F from $\mathcal{N}^{t,n}_{\Delta n}$, the integrality gap of any n^{δ} round Cutting Planes (or Th(1)) relaxation of L_G, the linear relaxation of the $2t$-CNF $G = \mathrm{Lift}^{\mathrm{T}}_{1,2}(F)$, is at least $1 - 1/2^{2t} + \epsilon$ with high probability.*

Proof. Given ϵ, fix $\Delta \gg 2^t \ln 2/\epsilon'^2$, where $(1 - 1/2^t + \epsilon')(2^t/(2^t-1) - \epsilon) = 1$. A random assignment satisfies each of F's clauses with probability $1-1/2^t$, so the expected number of satisfied clauses of F is $(1-1/2^t)\Delta n$. For appropriate choice of Δ, the probability that a random assignment satisfies more than a $1 - 1/2^t + \epsilon'$ fraction of equations is less than 2^{-n} by Chernoff bounds. Thus with high probability, no assignment satisfies more than a $1 - 1/2^t + \epsilon'$ fraction of F's equations. By the construction of G from F, each clause of F has precisely 2^t corresponding clauses in G. It follows that with high probability, no assignment satisfies more than a $1 - 1/2^{2t} + \epsilon$ fraction of G's clauses.

On the other hand, since $t \geq 3$, any Resolution refutation of F requires linear rank [10, 4]. Thus by Theorem 5.1 even after Th(1) inference of rank n^{δ} (and in particular n^{δ} rounds of Cutting Planes), there is some non-integral assignment α to the $x'_i s$ that satisfies all linear constraints corresponding to the clauses of G. Extend this assignment by setting all the z_i's to 1 and it follows that all constraints of L_G, are also satisfied. Thus we have a solution satisfying all equations that survives even after n^{δ} rounds. □

Note that since a random assignment on average satisfies a $1 - 1/2^{2t}$ fraction of clauses of any $2t$-CNF formula, this yields an optimal integrality gap for rank n^{δ} Th(1) inference for MAX-$2t$-SAT for any $t \geq 3$. Such a result was previously only known for the special case of Cutting Planes proofs [5] and the proof relied on the specific form of inference rather than the general sound inference allowed for Th(1) proofs.

6. DISCUSSION

We can obtain a similar integrality gap to that of Corollary 5.3 for any problem for which we can prove a decision tree lower bound. That is, take any optimization problem that can be expressed naturally as a t-CNF formula, and such that an integrality gap of $1-\gamma$ can be proven for decision trees. (Such a result is usually elementary to obtain.) Then by our lifting technique, we can show that any small $R^{cc}(2,C)$ refutation (including Cutting Planes and Th(1) proofs) for the lifted version has an integrality gap of $1-\gamma/2^t$. Our approach only works at present for proof systems that correspond to 2-player communication complexity. However, an extension of Theorem 5.2 to the multiparty setting (as was done with the qualitatively weaker Theorem 3.2 which was originally proven for the 2-player case [34]) would immediately yield integrality gaps for stronger matrix cut systems, such as LS+ and Lasserre.

Our work raises several other natural open questions. Our lower bounds apply for k up to $(1-o(1))\log\log n$. We conjecture that it should be possible to derive hardness escalation results that work when k is $\Omega(\log n)$. Can our methods be strengthened to obtain general (dag-like) size lower bounds? Such a result, even for $k=3$, would give the first unrestricted size lower bounds for Lovász-Schrijver proofs.

Finally, there are many general questions related to hardness escalation. What relationships are there between the various forms of hardness amplification, hardness escalation, hardness condensing, and hardness amplification? What other examples of hardness escalation can be shown, possibly given reasonable assumptions? It would be very interesting to obtain a hardness escalation result that lifts lower bounds for a circuit class where cryptography is not possible to a circuit class were cryptography is possible (e.g., lifting from DNF lower bounds to TC_0 lower bounds) as such a result would cross the "natural proof" barrier.

Acknowledgements

We thank Ran Raz and Rahul Santhanam for very helpful conversations. In particular we thank Rahul for suggesting the term *hardness escalation*.

7. REFERENCES

[1] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *JACM*, 48(4):778–797, 2001.

[2] P. Beame, T. Pitassi, and N. Segerlind. Lower bounds for Lovász-Schrijver systems and beyond follow from multiparty communication complexity. *SIAM J. Comput.*, 37(3):845–869, 2007.

[3] P. Beame and D.-T. Huynh-Ngoc. Multiparty communication complexity and threshold circuit size of AC^0. In *50th IEEE FOCS*, 53–62, 2009.

[4] E. Ben-Sasson and A. Wigderson. Short proofs are narrow – resolution made simple. *JACM*, 48(2):149–169, 2001.

[5] J. Buresh-Oppenheim, N. Galesi, S. Hoory, A. Magen, and T. Pitassi. Rank bounds and integrality gaps for cutting planes procedures. In *44th IEEE FOCS*, 318–327, 2003.

[6] J. Buresh-Oppenheim and R. Santhanam. Making hard problems harder. TR06-03, ECCC, 2006.

[7] M. Charikar, K. Makarychev, and Y. Makarychev. Integrality gaps for Sherali-Adams relaxations. In *41st ACM STOC*, 283–292, 2009.

[8] A. Chattopadhyay and A. Ada. Multiparty communication complexity of disjointness. TR08-002, ECCC, 2008.

[9] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4:305–337, 1973.

[10] V. Chvátal and Endre Szemerédi. Many hard examples for resolution. *JACM*, 35(4):759–768, 1988.

[11] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem proving. *CACM*, 5:394–397, 1962.

[12] J. Edmonds, R. Impagliazzo, S. Rudich, and J. Sgall. Communication complexity towards lower bounds on circuit depth. *Computational Complexity*, 10:210–246, 2001.

[13] K. Georgiou, A. Magen, T. Pitassi, and I. Tourlakis. Integrality gaps of $2-o(1)$ for vertex cover SDPs in the Lovasz-Schrijver hierarchy. In *48th IEEE FOCS*, 702–712, 2007.

[14] R.E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bull. of the AMS*, 64:275–278, 1958.

[15] D. Grigoriev. Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity. *TCS*, 259:613–622, 2001.

[16] D. Grigoriev, E. A. Hirsch, and D. V. Pasechnik. Complexity of semi-algebraic proofs. In *STACS* LNCS v. 2285, 419–430, 2002.

[17] R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *JCSS*, 67:367–375, 2001.

[18] R. Impagliazzo, T. Pitassi, and A. Urquhart. Upper and lower bounds on tree-like cutting planes proofs. In *9th IEEE LICS*, 220–228, 1994.

[19] M. Karchmer, R. Raz, and A. Wigderson. Super-logarithmic depth lower bounds via direct sum in communication complexity. *Computational Complexity*, 5:191–204, 1995.

[20] A. Kojevnikov and A. Itsykson. Lower bounds of static Lovasz-Schrijver calculus proofs for Tseitin tautologies. In *33rd ICALP*, 323–334, 2006.

[21] J. Krajíček. Discretely ordered modules as a first-order extension of the cutting planes proof system. *JSL*, 63(4):1582–1596, 1998.

[22] E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, 1997.

[23] J.B. Lasserre. An explicit SDP relaxation for nonlinear 0-1 programs. In *IPCO, LNCS v. 2081*, 293–303, 2001.

[24] T. Lee and A. Shraibman. Disjointness is hard in the multi-party number-on-the-forehead model. In *23rd IEEE CCC*, 81–91, 2008.

[25] L. Lovasz and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optimization*, 1(2):166–190, 1991.

[26] A. Maciel and T. Pitassi. Lower bounds on constant-depth Frege proofs with mod connectives modulo a hardness conjecture. In *11th IEEE LICS*, 189–200, 2006.

[27] S. Muroga. *Threshold logic and its applications.* John Wiley & Sons, 1971.

[28] N. Nisan and M. Szegedy. On the degree of boolean functions as real polynomials. *Computational Complexity*, 4:301–314, 1994.

[29] R. Raz and P. McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, 1999.

[30] G. Schoenebeck. Linear level Lasserre lower bounds for certain k-CSPs. In *49th IEEE FOCS*, 593–602, 2008.

[31] G. Schoenebeck, L. Trevisan, and M. Tulsiani. A linear round lower bound for Lovasz-Schrijver SDP relaxations of Vertex Cover. In *22nd IEEE CCC*, 205–216, 2007.

[32] N. Segerlind and T. Pitassi. Exponential lower bounds and integrality gaps for tree-like Lovasz-Schrijver procedures. In *ACM-SIAM SODA*, 355–364, 2009.

[33] A. Sherali and W. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Disc. Math.*, 3:411–430, 1990.

[34] A. A. Sherstov. Separating AC^0 from depth-2 majority circuits. In *39th ACM STOC*, 294–301, 2007.

[35] A. A. Sherstov. The pattern matrix method for lower bounds on quantum communication. In *40th ACM STOC*, 85–94, 2008. Full version: *CoRR, abs/0906.4291.*

[36] M. Tulsiani. CSP gaps and reductions in the Lasserre hierarchy. In *41st ACM STOC*, 303–312, 2009.