Multiparty Communication Complexity of Collision-Finding

Paul Beame* Computer Science & Engineering University of Washington

Michael Whitmeyer* Computer Science & Engineering University of Washington

November 13, 2024

Abstract

We prove an $\Omega(n^{1-1/k}\log k\ /2^k)$ lower bound on the k-party number-in-hand communication complexity of collision-finding. This implies a $2^{n^{1-o(1)}}$ lower bound on the size of tree-like cutting-planes proofs of the bit pigeonhole principle, a compact and natural propositional encoding of the pigeonhole principle, improving on the best previous lower bound of $2^{\Omega(\sqrt{n})}$.

1 Introduction

The pigeonhole principle asserting that there is no injective function $f : [m] \to [n]$ for $m > n$ is a cornerstone problem in the study of proof complexity. It is typically encoded as unsatisfiable conjunctive form formula (CNF), henceforth denoted PHP_n^m, on the variables $y_{i,j}$, each of which is an indicator that "pigeon" i is mapped to "hole" j.

It is well known that any refutation of PHP_n^{n+1} using resolution proofs requires size $2^{\Omega(n)}$ [Hak85] and the same asymptotic bound holds for all m that are $O(n)$ [BT88]. On the other hand, if we allow our proof system to reason about linear inequalities (for example using cutting-planes proofs), then it is easy to see that refuting PHP_n^{n+1} becomes easy – indeed, there exist polynomial size refutations of PHP_n^{n+1}.

Despite the pigeonhole principle having short cutting-planes refutations, the related clique-coloring formulas, which state that a graph cannot have both k-cliques and $k-1$-colorings, requires exponential-size cutting-planes refutation [Pud97].[1] The clique-coloring formula can be viewed as a kind of indirect pigeonhole principle: The k nodes of the clique correspond to the pigeons and $k-1$ colors correspond to the holes, but the representation of possible mappings is quite indirect.

It is natural to wonder about the extent to which indirection is required for the pigeonhole principle to be hard for cutting-planes reasoning. As part of studying techniques for cutting-planes proofs, Hrubeš and Pudlák [HP17] considered a very natural compact and direct way of expressing the pigeonhole principle, known as the *bit* or *binary* pigeonhole principle[2]. The bit pigeonhole principle analog of PHP_n^m (henceforth denoted BPHP_n^m) has $m\log n$ variables $x_{i,j}$ for $i \in [m], j \in [\log n]$ and the principle asserts that, when we organize these variables as an $m \times [\log n]$ matrix, the rows of the matrix all have distinct values. BPHP_n^m is the following CNF formula: for each $i \neq j \in [m]$, include the clauses of a CNF encoding that $x_i \neq x_j$. One can achieve this by including a clause for each $\alpha \in \{0,1\}^{\log n}$ expressing that $x_i \neq \alpha \vee x_j \neq \alpha$. The end result is a CNF with $\binom{m}{2}n$ clauses of size $2\log n$.

Using techniques related to those of [Pud97], Hrubeš and Pudlák [HP17] showed that BPHP_n^m requires cutting-planes refutations of size $2^{\Omega(n^{1/8})}$ for any $m > n$, proving that even a very direct representation of the pigeonhole principle is hard for cutting-planes proofs. Their arguments, like those of Pudlák, also apply to any proof system that has proof lines consisting of integer linear inequalities with two antecedents per inference that are sound with respect to 01-valued variables; such proofs are known alternatively as *semantic* cutting-planes proofs or $\mathsf{Th}(1)$ proofs [BPS07].

*Research supported by NSF grant CCF-2006359

[1] Lower bounds for restricted cutting-planes refutations of these formulas were earlier shown in [IPU94; BPR95]

[2] This encoding of the pigeonhole principle was introduced in [AMO15].

Recently, Dantchev, Galesi, Ghani, and Martin [Dan+24] exhibited a $2^{\Omega(n/\log n)}$ lower bound on the size of any general resolution refutation of BPHP_n^m for all $m > n$. In fact, they showed that BPHP_n^m requires proofs of size $2^{\Omega(n^{1-\varepsilon})}$ for a more powerful class of proof systems that extend resolution by operating on k-DNFs (known as $\mathrm{Res}(k)$ proofs) for $k \leq \log^{1/2-\varepsilon'} n$. (Note that any sound proof system operating on DNFs requires size at least $2^{n^{\Omega(1)}}$ to refute of PHP_n^{n+1} [PBI93; KPW95; Hås23].) In addition, [Dan+24] showed that BPHP_n^m has no refutations in the Sherali-Adams proof system [SA90] of size smaller than $2^{\Omega(n/\log^2 n)}$. Finally, just as PHP_n^m has polynomial-size Sum-of-Squares refutations [GHP01], Dantchev et al. showed that BPHP_n^m has polynomial-sized Sum-of-Squares refutations.

Given the large lower bounds for resolution, $\mathrm{Res}(k)$, and Sherali-Adams refutations of BPHP_n^m, it is natural to ask the extent to which the sub-exponential lower bounds can be improved for cutting-planes proofs; how close to a $2^{\Omega(n)}$ lower bound is possible? While the general question is still open, there has been progress towards this question for the restricted class of *tree-like* refutations. Tree-like proofs require that any time an inequality is used, it must be re-derived (i.e., the underlying graph of deductions is a tree); the polynomial-size cutting-planes refutations of PHP_n^{n+1} can be made tree-like. In contrast, Itsykson and Riazanov [IR20] showed that BPHP_n^m requires tree-like cutting-planes refutations of size $2^{\Omega(\sqrt{n})}$ when $m \leq n + \sqrt{n}$.

Our main result pushes this bound almost to its limit. Specifically, we prove that any tree-like semantic cutting-planes refutation of BPHP_n^m requires size $2^{n^{1-o(1)}}$ whenever $m \leq n + 2^{2\sqrt{\log n}-2}$.

In order to show this, we utilize a well-known connection between tree-like refutations and communication complexity. While the results of [IR20] for cutting planes relies on two-party communication complexity (and number-on-forehead multiparty communication for other results that we mention below), our stronger results are based on multiparty number-in-hand communication. In particular it is based on a similar natural collision-finding communication problem $\mathsf{Coll}_{m,\ell}^k$, in which each player $p \in [k]$ in the number-in-hand model receives an input in $x^{(p)} \in [\ell]^m$, and their goal is to communicate and find a pair $i \neq j \in [m]$ such that $x_i^{(p)} = x_j^{(p)}$ for all players $p \in [k]$. Such a communication problem is well-defined (in the sense that such a pair i, j exists) when $m > \ell^k$.

This collision-finding problem is intimately related to the unsatisfiable BPHP_n^m formula via the following natural search problem associated with any unsatisfiable CNF formula: Given unsatisfiable CNF φ, the associated search problem Search_φ takes as input a truth assignment α to the variables of φ and requires the output of the index of a clause of φ that is falsified by α. In particular the connection follows by considering a natural k-party number-in-hand communication game that we denote by $\mathrm{Search}_\varphi^k$ wherein the assignment α to the variables of φ is evenly distributed among the k players. and the players must communicate to find an an answer for $\mathrm{Search}_\varphi(\alpha)$.

It is not hard to see that if we have a communication protocol solving $\mathrm{Search}_{\mathrm{BPHP}_n^m}^k(\alpha)$ then such a protocol also solves $\mathsf{Coll}_{m,n^{1/k}}^k$ on input α. Our main technical result is a lower bound on $\mathsf{Coll}_{m,n^{1/k}}^k$ that holds even when we allow randomized protocols.

Theorem 1.1. *The randomized number-in-hand communication complexity of* $\mathsf{Coll}_{m,n^{1/k}}^k$ *is* $\Omega(n^{1-1/k}\log k\ /2^k)$ *whenever* $n+1 \leq m \leq n + 2^{k-2}n^{1/k}$.

We pause here to note that this bound is nearly tight. There is a deterministic protocol wherein the first player sends a subset of coordinates of size $\lceil m/n^{1/k} \rceil$ in which their inputs are all equal. This requires $\log \binom{m}{m/n^{1/k}} \lesssim (m/n^{1/k}) \log(m/n^{1/k})$ bits; when $m \approx n$, this is $O(n^{1-1/k}\log(n^{1/k})) = O(n^{1-1/k}\log n\ /k)$ bits of communication. Player two then announces a subset of these coordinates on which they are equal of size $\lceil m/n^{2/k} \rceil$. The players can continue in this manner until they have found a collision (which is guaranteed by the pigeonhole principle). Note that the amount of communication is handled by a geometric series, and is dominated by the first term, which results in communication $O(n^{1-1/k}\log n\ /k)$. This shows that up to logarithmic factors and a factor of 2^k, Theorem 1.1 is tight.

We state here a simplified corollary[3] of Theorem 1.1 which formalizes our lower bounds for cutting-planes refutations of BPHP_n^m.

[3] When m is somewhat larger, we can obtain somewhat weaker lower bounds.

Theorem 1.2. *Any tree-like semantic cutting-planes refutation of* BPHP_n^m *requires size* $2^{n^{1-2/\sqrt{\log n}-o(1/\sqrt{\log n})}}$ *when* $m \leq n + 2^{2\sqrt{\log n}-2}$.

We remark that Itsykson and Riazanov [IR20] utilized the same connection between communication and proof complexity to achieve their results. They were also interested in a k-party number-on-forehead version of $\mathsf{Coll}_{m,\ell}^k$ (in particular, in their version, the matrices are added rather than concatenated), which leads to weaker lower bounds in stronger proof systems $\mathsf{Th}(k-1)$ that manipulate degree $k-1$ polynomial inequalities.

Itsykson and Riazanov also left as an open problem whether their bounds for $\mathrm{Search}_{\mathrm{BPHP}_n^m}$ could be extended to the regime of the"weak" pigeonhole principle when $m = n + \Omega(n)$. Göös and Jain [GJ22] first answered this in the affirmative, giving an $\Omega(n^{1/12})$ lower bound on the randomized communication complexity of $\mathsf{Coll}_{2n,n^{1/2}}^2$. Yang and Zhang [YZ23] subsequently improved this to an $\Omega(n^{1/4})$ bound, which is tight for randomized computation. On the other hand, the results of Hrubeš and Pudlák [HP17] imply a size lower bound for *all* $m > n$ of $2^{\Omega(n^{1/8})}$ for the two-party deterministic *DAG-like* communication complexity of $\mathrm{Search}_{\mathrm{BPHP}_n^m}^2$, which is an incomparable model.

2 Preliminaries

2.1 Proof Complexity

Given an unsatisfiable formula φ, the field of proof complexity studies how long refutations need to be as a function of the size of φ. The length of a refutation in general depends on the allowable structure (lines, derivation rules, etc.) of a proof. In general, a proof *system* corresponds to a verifier that can check proofs of a certain format.

For most proof systems, a sequence of deductions can be thought of as a directed graph, where two (or possibly more) lines (whether given or derived) are combined soundly to create a new line. The underlying graph then has edges pointing from the derived inequality to its antecedents.[4] We say that a proof is *tree-like* if every inequality is used as an antecedent at most once in the proof – that is, if we want to use an inequality twice, we must derive it twice.

For example, we could define the lines in a proof system to be formulas, and allow the basic and common rule known as "resolution", which allows derivations of the following form:

$$(A \vee x) \wedge (B \vee \neg x) \implies A \wedge B,$$

where A and B are arbitrary formulas. As mentioned in the introduction, this is an extremely well-studied proof system in which it is well known that PHP_n^{n+1} requires exponentially long proofs.

A more powerful[5] proof system is the cutting planes proof system, denoted CP. For cutting planes proofs, lines are linear inequalities. We pause here to note that formulas can be trivially converted into linear inequalities; for example, $x \vee y \vee z$ could be converted to the inequality $x + y + z \geq 1$.

More generally, suppose we have a system of inequalities $Ax \leq b$ where A is a matrix with integer entries. We say that the system is unsatisfiable if it is unsatisfiable for any $x \in \{0,1\}^n$. The most basic form of the cutting planes proof system consists of three rules: allowing for addition of inequalities, allowing for multiplication of inequalities by positive integers, and most crucially, the rounded division rule, also known as the Gomory-Chvátal rule. The rounded division rule is the simple observation that if $a_1, \ldots, a_n$ are all integers with a common factor c, and we have the inequality $a^T x \leq b$, then we can derive that $\frac{1}{c} a^T x \leq \lfloor \frac{b}{c} \rfloor$. The floor here is crucial, and the only thing that allows for nontrivial deductions in this proof system.

In general, there are many more sound derivation rules for integer/linear inequalities than just rounded division (such as saturation [GNY19], just to name one), and even more generally, one may allow *any* sound derivation rule for linear inequalities which is known as *semantic* cutting planes or $\mathsf{Th}(1)$; we use the two interchangeably.

There is a well known connection between communication complexity and tree-like proofs, which we will now detail. Given any unsatisfiable formula φ, an assignment α to the variables can be distributed among

[4] The direction of arrows in this digraph may seem counterintuitive, but it is convenient when thinking of the graph as a search problem for a violated axiom. In this case, we can follow a path in the graph to find such a a violated axiom on one of the leaves.

[5] It turns out that cutting planes proofs can simulate resolution proofs, see e.g. [Juk14; RY20].

k players, who must then communicate in order to find a a violated clause in φ. This is a search problem, is denoted $\mathrm{Search}_\varphi(\alpha)$. Short tree-like proofs of the unsatisfiability of φ can often be converted into short protocols for Search_φ using standard techniques.

For example, a short tree-like proof of unsatisfiability of φ using the resolution rule naturally corresponds to a decision tree for finding a violated clause of φ in the following way. Every time the derivation of the form $(A \vee x) \wedge (B \vee \neg x) \implies (A \vee B)$ is made, we query x in order see whether A or B is necessarily false. We can continue in this manner, from the root of the tree-like refutation, until we hit an unsatisfied clause in the original formula.

On the other hand, tree-like semantic cutting planes refutations naturally correspond to *threshold* decision trees, which we now define.

Definition 2.1 (Threshold Decision Tree)**.** *A* threshold decision tree *is a tree whose vertices are labeled with inequalities of the form*

$$a_1x_1 + \cdots a_nx_n \leq b$$

where $a_1, \dots, a_n, b$ are integers. Edges are labelled with 0 or 1, and leaves are axioms of a system of inequalities $Ax \leq b$.

We traverse a threshold decision tree by computing the threshold function at the root, following the corresponding edge, and continuing in this manner until we hit a leaf. We say that a threshold decision tree computes the search problem for a formula φ if this process leads to a leaf corresponding to a violated clause in φ.

First, we have the following well-known lemma, which states that one can derive a low-depth threshold decision tree from a small $\mathsf{Th}(1)$ refutation.[6]

Proposition 2.2. *Given a size S tree-like $\mathsf{Th}(1)$ refutation of an unsatisfiable system $Ax \leq b$, there is a depth $O(\log S)$ threshold decision tree finding a violated axiom.*

Proposition 2.2 goes back to the work of Impagliazzo, Pitassi, and Urquhart [IPU94], and can also be found for instance in [Juk14]. We omit the proof, but the idea is a common one: find a node in the tree with roughly half (between 1/3 and 2/3) of the leaves as its descendants, and make that the root of the threshold decision tree, and recurse.

2.2 Communication Complexity

We mainly focus on k-party number-in-hand communication, wherein each player p receives an input $x^{(p)}$, and the players' goal is to communicate as little as possible in order to compute a known function or relation involving their collective inputs. In general, players may have access to shared randomness, and we allow incorrect answers with probability 1/3.

An important function for us is the number-in-hand disjointness problem with k players and input size n, henceforth denoted DISJ_n^k. This is the communication problem wherein each player and input in $\{0,1\}^n$, and they must decide if there exists a coordinate i for which they all have a 1 in that coordinate. Disjointness in general is an extremely well-studied problem [CP10; RY20], and for the specific case of the NIH model, we have the following lower bound due to Braverman and Oshman.

Theorem 2.3 ([BO15])**.** *The randomized communication complexity of DISJ_n^k is $\Omega(n \log k)$.*

Our results rely on the following connection between threshold decision trees for finding violated clauses and k-party NIH communication. It is closely related to previous work.

Lemma 2.4. *For $x \in \{0,1\}^n$, if an unsatisfiable system $Ax \leq b$ on has a threshold decision tree of depth $d \leq n$ finding a violated axiom, then for any partition of the input variables into k parts there is a randomized protocol for $\mathrm{Search}_{NIH}^k(Ax \leq b)$ using $O(dk \log k \log n)$ bits of communication.*

[6] As noted in [Juk14], there is no meaningful converse to this statement, since if there are m inequalities in our unsatisfiable system, there exists a trivial depth m threshold decision tree finding one that is violated.

Lemma 2.4 is similar for example to Lemma 5 in [IPU94] or Lemma 19.11 in [Juk14], but is slightly stronger, so we include the proof.

The idea is that the players can use the given threshold decision tree to construct a protocol. Without loss of generality, using a well-known theorem of Muroga [MTT61] without communication the players can replace each of the inequalities in the decision tree with an equivalent one over the Boolean hypercube with coefficients that are not too large.

The players then start at the root and evaluate each associated inequality using an efficient randomized protocol to evaluate each threshold function with high probability and moving to the appropriate child node until it reaches a leaf. Such a protocol dates back to Nisan [Nis93] and was tightened by Viola [Vio15].

We first state the results of Muroga and Viola required to formalize this construction.

Proposition 2.5 ([Mur71; MTT61])**.** *Consider a threshold function $f : \{0,1\}^n \to \{0,1\}$ of the form*

$$f(x) = \begin{cases} 1 & \text{if } a_1x_1 + \cdots + a_nx_n \leq b \\ 0 & o.w. \end{cases}$$

Then f is equivalent to another threshold function

$$f'(x) = \begin{cases} 1 & \text{if } a'_1x_1 + \cdots + a'_nx_n \leq b' \\ 0 & o.w. \end{cases}$$

where each $a'_1, \ldots a'_n, b'$ is at most $2^{-n}(n+1)^{(n+1)/2}$ in magntiude.

In particular, Proposition 2.5 implies that we may assume in a $\mathsf{Th}(1)$ proof that, up to a factor of two in the size, every derived inequality has coefficients of magnitude at most $O(n^n)$.

Proposition 2.6 ([Vio15])**.** *Suppose that each player $p \in [k]$ receives an input $x^{(p)} \in [-2^n, 2^n]$. Then there is a randomized number-in-hand protocol with error at most ε that determines whether $\sum_p x^{(p)} > s$ and communicates $O(k \log k \log(n/\varepsilon))$ bits.*

Corollary 2.7. *Suppose that each player $p \in [k]$ receives an input $x^{(p)} \in [2^t]$. Then they can execute a randomized number-in-hand protocol to determine whether $\sum_p a_p x^{(p)} \leq b$, where each $|a_p| \leq 2^w$ with error at most ε using at most $O(k \log k \log((w+t)/\varepsilon))$ bits of communication.*

We are now ready to prove Lemma 2.4.

Proof of Lemma 2.4. Given a threshold decision tree of depth d, we simply traverse it from the root. For each inequality, if the magnitudes of its weights are not bounded by $2^{O(n \log n)}$, then we replace it with an equivalent threshold function whose weights are bounded using Proposition 2.5. Then, using Corollary 2.7, the players communicate $O(k \log((n \log n)/\varepsilon) \log k)$ to compute the threshold function with error probability ε. Setting $\varepsilon = \Theta(1/d)$ and continuing in this manner, by a union bound the threshold functions are all computed correctly with constant probability.

Using the assumption that $d \leq n$, this yields a protocol communicating $O(dk \log k \log n)$ bits. ☐

Given a system of unsatisfiable inequalities $Ax \leq b$, and a partition of an assignment $\alpha \in \{0,1\}^n$ between k players, there is a natural (number-in-hand) communication game wherein players must communicate to a find an axiom violated by α. Lemma 2.4 implies the following result connecting communication complexity and proof complexity.

Lemma 2.8. *For any partition of n variables into k parts, if $\mathrm{Search}^k_{NIH}(Ax \leq b)$ requires t bits of communication, then any tree-like $\mathsf{Th}(1)$ refutation of $Ax \leq b$ requires size $2^{\Omega(t/(k \log k \log n))}$.*

Proof. By Proposition 2.2, given a size S tree-like $\mathsf{Th}(1)$ proof, we get a depth $d = O(\log S)$ threshold decision tree finding a violated axiom. By Lemma 2.4, there exists a communication protocol finding a violated axiom using $O(\log S \ k \log k \ \log n)$ bits of communication. This implies that $\log(S) k \log k \log n \geq ct$ for constant c, which in turn implies S is $2^{\Omega(t/(k \log k \ \log n))}$, as desired. ☐

3 Communication Lower Bound

In this section we prove Theorem 1.1, which we now recall.

Theorem. *The randomized number-in-hand communication complexity of* $\mathsf{Coll}^k_{m,n^{1/k}}$ *is at least* $\Omega(n^{1-1/k}2^{-k}\log k)$ *whenever* $n+1 \le m \le n + 2^{k-2}n^{1/k}$.

The idea for the proof is to exhibit a random reduction from the decision problem $\mathsf{DISJ}^k_{m^{k-1}}$ to the collision problem. This is analogous to the approach of Itsykson and Riazonov [IR20] for number-on-forehead communication complexity which used lower bounds for disjointness in that model and a randomized decision-to-search reduction paradigm introduced by Raz and Wigderson [RW92] to prove lower bounds on the monotone depth complexity of matching. The details and parameters of our reduction are necessarily quite different.

In our setting, we embed k players' inputs to a disjointness problem into k matrices such that, when these matrices are concatenated, the resulting matrix has distinct rows if and only if the players' inputs were disjoint. We can then add a few "fake" rows to this matrix and run our algorithm for $\mathsf{Coll}^k_{m,n^{1/k}}$, and see if the collisions it finds involve the fake rows or not. If so, we conclude that the inputs were disjoint and if not, we know that they were not disjoint.

The following key combinatorial lemma allows us to carry out the first step of this process.

Lemma 3.1. *For all integers* $k \ge 1$ *there exist matrices* $M^1_k \in \{0,1\}^{2^k \times k}$ *and* $M^0_k \in \{0,1\}^{2^k \times k}$ *such that*

1. M^1_k *has* 2^{k-1} *unique pairs of identical rows.*
2. *For any string* $b \in \{0,1\}^k$*, define the matrix* $M_k(b_1, \dots b_k)$ *as the matrix formed by making its* i*-th column equal to the* i*-th column of* M^0_k *if* $b_i = 0$*, and equal to the* i*-th column of* M^1_k *if* $b_i = 1$*. Then* $M_k(b)$ *has unique rows for all* $b \neq \vec{1}$*.*

We defer the proof of Lemma 3.1 to Section 3.1.

Proof of Theorem 1.1 using Lemma 3.1. As alluded to, we will reduce the NIH disjointness problem to our bit-pigeonhole problem. Namely, we will reduce $\mathsf{DISJ}^k_{m^{k-1}}$ to a $\mathsf{Coll}^k_{\widetilde{m},\widetilde{\ell}}$ communication game, where $\widetilde{m} = m^k 2^k + 2^{k-1}m$, and $\widetilde{\ell} = 2m$.

The players get $x^{(1)}, \dots, x^{(k)} \subseteq [m^{k-1}]$ (viewed as bit strings of length m^{k-1}), and need to determine whether $x^{(1)} \cap \cdots \cap x^{(k)} = \emptyset$.

First, we define

$$B_{m^k}(j) := \begin{bmatrix} \mathrm{bin}_{m^k}(j) \\ \vdots \\ \mathrm{bin}_{m^k}(j) \end{bmatrix} \in \{0,1\}^{2^k \times k \log m},$$

where we have repeated the same row 2^k times.

For each $i \in [m^{k-1}]$, consider the matrix $M_k(x^{(1)}_i, \dots, x^{(k)}_i)$ from Lemma 3.1. Note that each player p knows the p-th column of $M_k(x^{(1)}_i, \dots, x^{(k)}_i)$ for all i, but without communication the other players do not know this column.

Then we can define

$$\widetilde{M} := \begin{bmatrix} M_k(x_1^{(1)}, \ldots, x_1^{(k)}) & B_{m^k}(0) \\ M_k(x_1^{(1)}, \ldots, x_1^{(k)}) & B_{m^k}(1) \\ \vdots & \vdots \\ M_k(x_1^{(1)}, \ldots, x_1^{(k)}) & B_{m^k}(m-1) \\ M_k(x_2^{(1)}, \ldots, x_2^{(k)}) & B_{m^k}(m) \\ M_k(x_2^{(1)}, \ldots, x_2^{(k)}) & B_{m^k}(m+1) \\ \vdots & \vdots \\ M_k(x_2^{(1)}, \ldots, x_2^{(k)}) & B_{m^k}(2m-1) \\ \vdots & \vdots \\ \vdots & \vdots \\ M_k(x_{m^{k-1}}^{(1)}, \ldots, x_{m^{k-1}}^{(k)}) & B_{m^k}(m^k-1) \end{bmatrix} \in \{0,1\}^{m^k \cdot 2^k \times (k \log m + k)}.$$

Observe that each player p can construct their "part" of this matrix without communicating by constructing the p-th column of every $M_k^{S_i}$ matrix (which only depends on $x^{(p)}$), and then taking the the p-th part of each of the $B_{m^k}(j)$ matrices.

Lemma 3.1 lets us connect the distinctness property of this matrix with the disjointness property of the players' inputs.

Claim 3.2. *If $(x^{(1)}, \ldots, x^{(k)})$ are disjoint, then $\widetilde{M}$ has distinct rows.*

Proof. The only possible collisions happen in every group of 2^k rows, since $B_{m^k}(j)$ has every row different from $B_{m^k}(i)$ for all $i \neq j$. Within these groups, by Lemma 3.1, if the inputs are disjoint then there are no collisions. □

Claim 3.3. *If X is not disjoint, then there are at least $2^{k-1}m$ pairs of colliding rows in $\widetilde{M}$.*

Proof. Any coordinate i for which $x_i^{(p)} = 1$ for all p generates $S_i = \emptyset$, which by Lemma 3.1 generates $2^{k-1}m$ such pairs, since input i was repeated m times. □

We cannot run any collision protocol for $\widetilde{M}$ yet, as there are not guaranteed collisions. To address this, the players use shared randomness to put an additional $2^{k-1}m$ rows at the bottom of $\widetilde{M}$. These rows will be chosen randomly with the following two properties:

1. Each fake row will be distinct.
2. Each player's "part" of the matrix (which consists of $\log m + 1$ columns) when restricted to these rows will repeat the $2m$ unique possible bit strings an addition $2^{k-1}m/(2m) = 2^{k-2}$ times.[7]

Denote this new matrix M. M now has "fake" collisions which involve any of the last $2^{k-1}m$ rows.

Let $\mathcal{A}$ denote the randomized protocol solving the $\mathsf{Coll}^k_{(2m)^k+2^{k-1}m,2m}$ problem.

Observe that if the inputs are disjoint, then the only collisions in M involve fake rows. The players would like to feed their parts of this matrix into $\mathcal{A}$ and conclude that their inputs are disjoint if the output involves a fake row, and conclude that they were not disjoint if the output involves two non-fake rows. However, this is problematic, as we have no guarantees over how $\mathcal{A}$ behaves, and it could always find a collision involving one of the last $2m$ rows (which it knows are fake), regardless of if there are other collisions.

This necessitates the following random shuffling.

1. Each player applies (the same) random permutation $\pi : [2^k m^k + 2^{k-1}m] \to [2^k m^k + 2^{k-1}m]$ which shuffles the rows of M.
2. Each player applies an individual random permutation $\pi^{(p)} : [2m] \to [2m]$ to each of their rows. Note that this preserves collisions/distinctness in the concatenation.

[7] This is important because if we bias and have a certain fake row appear more often in the input for player p, then $\mathcal{A}$ could potentially detect and use this to its advantage.

Denote $\vec{\pi} := (\pi, \pi^{(1)}, \ldots, \pi^{(k)})$, and call this final matrix $M_{\vec{\pi}}$.

Algorithm: The algorithm for disjointness is as follows: the players use their inputs and shared randomness to compute (without communication) their respective parts of 5 independent copies of an $M_{\vec{\pi}}$ constructed in the above manner, and run $\mathcal{A}$ using these as inputs. The players then examine the outputs $(i_1, j_1), \ldots (i_5, j_5)$ of the algorithm on these five inputs. They then exchange an additional $O(k \log m)$ bits to determine if each claimed collision actually was a collision. Finally, if any of the claimed collisions were actually collisions on rows that were not fake (under the appropriate permutation), then the players can conclude with certainty that their inputs were not disjoint. Otherwise, if $\mathcal{A}$ only ever finds colliding pairs that involve a row the players know is fake (or otherwise fail to find any collisions), then players guess that $(x^{(1)}, \ldots, x^{(k)})$ were disjoint.

Analysis: We analyze one iteration of the algorithm. Suppose $\mathcal{A}$ has error probability at most $1/3$ – that is, with probability at least $2/3$, it outputs two rows i, j that are equal.

Suppose $(x^{(1)}, \ldots, x^{(k)})$ are disjoint. Then by assumption, $\mathcal{A}$ finds a collision with probability at least $2/3$, and we know this collision will always involve a fake row by Claim 3.2. Therefore the players will correctly output that their inputs were distinct with probability at least $2/3$, and this is only improved by the five-fold repetition.

Otherwise, suppose that the players' inputs were not disjoint. Suppose further that $\mathcal{A}$ successfully finds a collision – this happens with probability at least $2/3$. Recall that by Claim 3.3 $M_{\vec{\pi}}$ will have at least $2^{k-1}m$ distinct pairs of real collisions. Adding the $2^{k-1}m$ fake rows produced additional "fake" collisions. These fake rows could have created up to $2^{k-1}m$ additional unique pairs of fake collisions, or could have "joined" the real collisions, creating up to $2^{k-1}m$ groups of 3 equal rows in $M_{\vec{\pi}}$.

If $\mathcal{A}$ outputs a collision from one of the groups of three, then because we applied random permutations to the rows, it is equally likely to have chosen any of the 3 possible pairs. Therefore, with probability at least $1/3$, it outputs a real collision, and the players successfully discover that they are not disjoint. Otherwise, if $\mathcal{A}$ outputs one of the unique collision pairs, then (again because we have applied random permutations to the rows), any such unique collision is equally likely to be output. If t of the fake rows formed a group of three with real collisions, then that leaves at most $2^{k-1}m - t$ fake rows to collide with a different unique row. It also leaves $2^{k-1}m - t$ untouched real collisions, so $\mathcal{A}$ outputs a real collision with probability at least $1/2$. Either way, the probability that $\mathcal{A}$ outputs a real collision is at least $2/3 \cdot 1/3 = 2/9$.

Therefore, after repeating this 5 times independently, the probability of seeing at least one real collision is at least $1 - (7/9)^5 > 2/3$.

Let $n := (2m)^k$. We have shown that if $\mathsf{Coll}^k_{\widetilde{m},\widetilde{\ell}}$ with input size $\widetilde{m} = 2^k m^k + 2^{k-1}m = n + 2^{k-2}n^{1/k}$ can be solved with $o(n^{1-1/k} \log k / 2^k)$ communication, then we can solve the decision disjointness problem with $o(n^{1-1/k} \log k / 2^k) + O(k \log m)$ which is at most $o(m^{k-1} \log k)$, contradicting the $\Omega(m^{k-1} \log k)$ lower bound from Theorem 2.3. □

3.1 Proof of Lemma 3.1

We first recall Lemma 3.1.

Lemma. *For all integers $k \geq 1$ there exist matrices $M_k^1 \in \{0,1\}^{2^k \times k}$ and $M_k^0 \in \{0,1\}^{2^k \times k}$ such that*

1. *M_k^1 has 2^{k-1} unique pairs of identical rows.*
2. *For any string $b \in \{0,1\}^k$, define the matrix $M_k(b_1, \ldots b_k)$ as the matrix formed by making its i-th column equal to the i-th column of M_k^0 if $b_i = 0$, and equal to the i-th column of M_k^1 if $b_i = 1$. Then $M_k(b)$ has unique rows for all $b \neq \vec{1}$.*

Proof. Let $\mathcal{E}_k \subseteq \{0, \ldots, k-1\}$ be the set of integers with an even number of 1s in their binary representation.

Define

$$M_k^1 = \begin{bmatrix} \text{bin}_k(y_1) \\ \text{bin}_k(y_1) \\ \text{bin}_k(y_2) \\ \text{bin}_k(y_2) \\ \vdots \\ \text{bin}_k(y_{2^{k-1}}) \\ \text{bin}_k(y_{2^{k-1}}) \end{bmatrix} \in \{0,1\}^{2^k \times k},$$

where y_i is the i-th smallest number in $\mathcal{E}_k$.

We pause here to note that each of the columns of M_k^1 are actually each the truth table of a linear function. Let $f_1 : \{0,1\}^k \to \{0,1\}$ be the linear function $f_1(x) = \langle x, e_1 \rangle$, where e_1 is the first standard basis vector. Then we can describe the first column of M_k^1 as the truth table of f_1. More generally, we have that for $i < k$ the i-th column of M_k^1 is the truth table of $f_i(x) = \langle x, e_i \rangle$, and the last column is the truth table of $f_k(x) = \langle x, e_1 + \cdots + e_{k-1} \rangle$.

If we define F_k^1 to be the matrix whose column i is the vector whose inner product we are taking with x in f_i, and $B_k \in \mathbb{F}_2^{2^k \times k}$ whose rows are the binary strings written in order, then we have that

$$M_k^1 = \begin{bmatrix} \text{bin}_k(0) \\ \text{bin}_k(1) \\ \vdots \\ \text{bin}_k(2^k - 1) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \cdots & 1 \\ 0 & 1 & 0 & \cdots & 1 \\ 0 & 0 & 1 & \ddots & 1 \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} =: B_k F_k^1$$

where all operations are over $\mathbb{F}_2$. M_k^1 has repeated rows precisely because F_k^1 has linearly dependent columns.

With this perspective in mind, if we can find a $k \times k$ matrix F_k^0 over $\mathbb{F}_2$ such that replacing any (nonzero) number of columns of F_k^1 with corresponding columns in F_k^0 produces a matrix with linearly independent columns, then we are done, as we can let $M_k^0 := B_k F_k^0$.

We define F_k^0 to be the following lower triangular matrix:

$$F_k^0 := \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix},$$

Clearly F_k^0 is full rank.

We claim that replacing any set nonempty set S of columns of F_k^1 with the corresponding columns in F_k^0 produces a matrix F_k^S with linearly independent columns. Consider arbitrary k, and arbitrary nonempty $S \subseteq [k]$.

Case 1: Suppose $k \in S$, that is, the last column of F_k^S is e_k. Then our matrix is lower triangular with 1s on the diagonal, and so has linearly independent columns.

Case 2: Suppose $k \notin S$, that is, the last column of F_k^S is $e_1 + \cdots + e_{k-1}$. In this case, the first $k-1$ columns are lower triangular with 1s on the diagonal, and therefore their span is equal to $\mathsf{span}\{e_1, \ldots, e_{k-1}\}$. It suffices then to show that also e_k is in the column span. Towards this goal, take the minimal $0 < i < k$ such that $i \in S$, and observe that summing the first i columns of F_k^S equals $\vec{1}$, the all 1s vector. Adding this to the last column produces e_k, so we are done. □

4 Proof Complexity Lower Bounds

In this section, we prove the following more detailed version of Theorem 1.2.

Theorem 4.1. *When $m \le n + 2^{k-2}n^{1/k}$, any tree-like semantic cuttings planes refutation of* BPHP_n^m *must have size at least* $2^{\Omega(n^{1-1/k}2^{-k}/(k\log n))}$.

Proof. The theorem follows quite readily from Lemma 2.8 and the fact that $\mathrm{Search}^k_{\mathrm{BPHP}_n^m}$ reduces to $\mathsf{Coll}^k_{m,n^{1/k}}$. If we translate BPHP_n^m to a system of inequalities, then there are $O(nm^2) = O(n^3)$ inequalities on $m \log n$ variables.

Lemma 2.8 then says that any tree-like semantic cutting planes proof of the unsatisfiability of BPHP_n^m must have size at least $2^{\Omega(t/(k\log k \ \log n))}$. By Theorem 1.1, $t = \Omega(n^{1-1/k}2^{-k}\log k)$. Plugging this in yields that the tree-like size must be at least $2^{\Omega(n^{1-1/k}2^{-k}/(k\log n))}$. □

From Theorem 4.1 we achieve Theorem 1.2, which we restate now.

Corollary. *Any tree-like semantic cutting planes refutation of* BPHP_n^m *requires size* $2^{n^{1-2/\sqrt{\log n}-o(1/\sqrt{\log n})}}$ *when* $m \le n + 2^{2\sqrt{\log n}-2}$.

Proof. Let $k = \sqrt{\log n}$. Plugging this into the bound from Theorem 4.1, we get that $m \le n + 2^{\sqrt{\log n}-2}n^{1/\sqrt{\log n}} = 2^{2\sqrt{\log n}-2}$. In this regime, Theorem 4.1 gives the size lower bound

$$2^{\Omega(n^{1-1/\sqrt{\log n}}n^{-1/\sqrt{\log n}}/(\log^{3/2} n))} = 2^{cn^{1-2/\sqrt{\log n}-1.5\log\log n/\log n}},$$

for appropriate constant c. Using the fact that $c = n^{\log c/\log n}$, the bound becomes

$$2^{n^{1-2/\sqrt{\log n}-o(1/\sqrt{\log n})}}.$$

□

5 Discussion and Future Work

We end by discussing some related problems and directions for future study. In particular, we highlight three possible directions.

1. **DAG-like communication lower bounds.** The lower bounds we prove are in the randomized communication model, and lead to lower bounds for tree-like cutting planes refutations.

   On the other hand, Hrubes and Pudlák's [HP17] work, as noted in the introduction, actually implies an $\Omega(n^{1/8})$ DAG-like deterministic communication lower bound for $\mathrm{Search}^2_{\mathrm{BPHP}_n^m}$ for arbitrary $m > n$ (i.e. even in the "weak" regime).

   It would be interesting to see if one can prove DAG-like communication lower bounds for the k-player analog $\mathrm{Search}^k_{\mathrm{BPHP}_n^m}$. We have shown that generalizing to k players helps in the tree-like case, and perhaps this holds true in the DAG-like setting as well. It may also be useful to first consider the strong setting when $m = n + 1$.

2. **Weak Bit Pigeonhole Principle**. Our focus in this work has been solely on the strong bit pigeonhole principle, when $m \le n + \sqrt{n}$ or even smaller.

   The results of Itsykson and Riazanov [IR20] also fall in this regime of parameters, and it was left as an open problem in their work whether any sort of lower bound on in the "weak" regime $m = n + \Omega(n)$ might hold.

   In this regime, when $k = 2$, better upper bounds are possible for $\mathrm{Search}_{\mathrm{BPHP}_n^{n+\Omega(n)}}$ – indeed, via the birthday paradox, there is a randomized protocol solving $\mathrm{Search}_{\mathrm{BPHP}_n^{n+\Omega(n)}}$ using only $O(n^{1/4}\log n)$ bits, which is essentially tight by the recent lower bound of Yang and Zhang [YZ23], who used ideas inspired from the lifting literature to prove their lower bound.

   We conjecture that by considering k-party number-in-hand communication model as we have done here should also yield stronger lower bounds in the weak regime. Using similar birthday paradox ideas to the $k = 2$ case, one can solve $\mathrm{Search}^k_{\mathrm{BPHP}_n^{n+\Omega(n)}}$ using $O(n^{1/2-1/2k}\log n)$ bits of communication; we conjecture that this is optimal.

3. Finally, we highlight that the loss of 2^k in the denominator of Theorem 1.1 could potentially be improved. Indeed we do not suspect that it should be present at all, and we conjecture that $\mathrm{Search}^k_{\mathrm{BPHP}^m_n}$ should remain hard for k all the way up to $\log n$. However, it seems unlikely that any reduction from k-party disjointness would be able to achieve this since an additional input bit per player seems essential in maintaining the conversion.

6 Acknowledgements

We thank Pavel Pudlák for discussions that led us to this research direction.

References

[AMO15] Albert Atserias, Moritz Müller, and Sergi Oliva. "Lower Bounds for DNF-Refutations of a Relativized Weak Pigeonhole Principle". In: *J. Symb. Log.* 80.2 (2015), pp. 450–476. DOI: 10.1017/JSL.2014.56. URL: https://doi.org/10.1017/jsl.2014.56.

[BO15] Mark Braverman and Rotem Oshman. "The Communication Complexity of Number-In-Hand Set Disjointness with No Promise". In: *Electron. Colloquium Comput. Complex.* TR15-002 (2015). ECCC: TR15-002. URL: https://eccc.weizmann.ac.il/report/2015/002.

[BPR95] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. "Lower bounds for cutting planes proofs with small coefficients". In: *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA*. ACM, 1995, pp. 575–584. DOI: 10.1145/225058.225275. URL: https://doi.org/10.1145/225058.225275.

[BPS07] Paul Beame, Toniann Pitassi, and Nathan Segerlind. "Lower Bounds for Lov[a-acute]sz–Schrijver Systems and Beyond Follow from Multiparty Communication Complexity". In: *SIAM J. Comput.* 37.3 (2007), pp. 845–869. DOI: 10.1137/060654645. URL: https://doi.org/10.1137/060654645.

[BT88] Samuel R. Buss and György Turán. "Resolution Proofs of Generalized Pigeonhole Principles". In: *Theor. Comput. Sci.* 62.3 (1988), pp. 311–317. DOI: 10.1016/0304-3975(88)90072-2. URL: https://doi.org/10.1016/0304-3975(88)90072-2.

[CP10] Arkadev Chattopadhyay and Toniann Pitassi. "The story of set disjointness". In: *SIGACT News* 41.3 (2010), pp. 59–85. DOI: 10.1145/1855118.1855133. URL: https://doi.org/10.1145/1855118.1855133.

[Dan+24] Stefan S. Dantchev, Nicola Galesi, Abdul Ghani, and Barnaby Martin. "Proof Complexity and the Binary Encoding of Combinatorial Principles". In: *SIAM J. Comput.* 53.3 (2024), pp. 764–802. DOI: 10.1137/20M134784X. URL: https://doi.org/10.1137/20m134784x.

[GHP01] Dima Grigoriev, Edward A. Hirsch, and Dmitrii V. Pasechnik. "Complexity of semi-algebraic proofs". In: *Electron. Colloquium Comput. Complex.* TR01-103 (2001). ECCC: TR01-103. URL: https://eccc.weizmann.ac.il/eccc-reports/2001/TR01-103/index.html.

[GJ22] Mika Göös and Siddhartha Jain. "Communication Complexity of Collision". In: *Electron. Colloquium Comput. Complex.* TR22-096 (2022). ECCC: TR22-096. URL: https://eccc.weizmann.ac.il/report/2022/096.

[GNY19] Stephan Gocht, Jakob Nordström, and Amir Yehudayoff. "On Division Versus Saturation in Pseudo-Boolean Solving". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019.* Ed. by Sarit Kraus. ijcai.org, 2019, pp. 1711–1718. DOI: 10.24963/IJCAI.2019/237. URL: https://doi.org/10.24963/ijcai.2019/237.

[Hak85] Armin Haken. "The Intractability of Resolution". In: *Theor. Comput. Sci.* 39 (1985), pp. 297–308. DOI: 10.1016/0304-3975(85)90144-6. URL: https://doi.org/10.1016/0304-3975(85)90144-6.

[Hås23] Johan Håstad. "On small-depth Frege proofs for PHP". In: *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023.* IEEE, 2023, pp. 37–49. DOI: 10.1109/FOCS57990.2023.00010. URL: https://doi.org/10.1109/FOCS57990.2023.00010.

[HP17] Pavel Hrubes and Pavel Pudlák. "Random formulas, monotone circuits, and interpolation". In: *Electron. Colloquium Comput. Complex.* TR17-042 (2017). ECCC: TR17-042. URL: https://eccc.weizmann.ac.il/report/2017/042.

[IPU94] Russell Impagliazzo, Toniann Pitassi, and Alasdair Urquhart. "Upper and Lower Bounds for Tree-Like Cutting Planes Proofs". In: *Proceedings of the Ninth Annual Symposium on Logic in Computer Science (LICS '94), Paris, France, July 4-7, 1994.* IEEE Computer Society, 1994, pp. 220–228. DOI: 10.1109/LICS.1994.316069. URL: https://doi.org/10.1109/LICS.1994.316069.

[IR20] Dmitry Itsykson and Artur Riazanov. "Proof complexity of natural formulas via communication arguments". In: *Electron. Colloquium Comput. Complex.* TR20-184 (2020). ECCC: TR20-184. URL: https://eccc.weizmann.ac.il/report/2020/184.

[Juk14] Stasys Jukna. "Boolean Function Complexity Advances and Frontiers". In: *Bull. EATCS* 113 (2014). URL: http://eatcs.org/beatcs/index.php/beatcs/article/view/275.

[KPW95] Jan Krajícek, Pavel Pudlák, and Alan R. Woods. "An Exponenetioal Lower Bound to the Size of Bounded Depth Frege Proofs of the Pigeonhole Principle". In: *Random Struct. Algorithms* 7.1 (1995), pp. 15–40. DOI: 10.1002/RSA.3240070103. URL: https://doi.org/10.1002/rsa.3240070103.

[MTT61] Saburo Muroga, Iwao Toda, and Satoru Takasu. "Theory of majority decision elements". In: *Journal of the Franklin Institute* 271.5 (1961), pp. 376–418.

[Mur71] Saburo Muroga. *Threshold logic and its applications*. Wiley, 1971. ISBN: 978-0-471-62530-8.

[Nis93] Noam Nisan. "The communication complexity of threshold gates". In: *Combinatorics, Paul Erdos is Eighty* 1.301-315 (1993), p. 6.

[PBI93] Toniann Pitassi, Paul Beame, and Russell Impagliazzo. "Exponential Lower Bounds for the Pigeonhole Principle". In: *Comput. Complex.* 3 (1993), pp. 97–140. DOI: 10.1007/BF01200117. URL: https://doi.org/10.1007/BF01200117.

[Pud97] Pavel Pudlák. "Lower Bounds for Resolution and Cutting Plane Proofs and Monotone Computations". In: *J. Symb. Log.* 62.3 (1997), pp. 981–998. DOI: 10.2307/2275583. URL: https://doi.org/10.2307/2275583.

[RW92] Ran Raz and Avi Wigderson. "Monotone Circuits for Matching Require Linear Depth". In: *J. ACM* 39.3 (1992), pp. 736–744. DOI: 10.1145/146637.146684. URL: https://doi.org/10.1145/146637.146684.

[RY20] Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020.

[SA90] Hanif D. Sherali and Warren P. Adams. "A Hierarchy of Relaxations Between the Continuous and Convex Hull Representations for Zero-One Programming Problems". In: *SIAM J. Discret. Math.* 3.3 (1990), pp. 411–430. DOI: 10.1137/0403036. URL: https://doi.org/10.1137/0403036.

[Vio15] Emanuele Viola. "The communication complexity of addition". In: *Combinatorica* 35 (2015), pp. 703–747.

[YZ23] Guangxu Yang and Jiapeng Zhang. "Communication Lower Bounds for Collision Problems via Density Increment Arguments". In: *Electron. Colloquium Comput. Complex.* TR23-159 (2023). ECCC: TR23-159. URL: https://eccc.weizmann.ac.il/report/2023/159.