Electronic Colloquium on Computational Complexity, Report No. 53 (2005)

Lower bounds for Lovász-Schrijver systems and beyond, using multiparty communication complexity

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

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

Nathan Segerlind[‡]
University of Washington
Seattle, WA 98195-2350
nsegerli@cs.washington.edu

May 3, 2005

Abstract

We prove that an $\omega(\log^3 n)$ lower bound for the three-party number-on-the-forehead (NOF) communication complexity of the set-disjointness function implies an $n^{\omega(1)}$ size lower bound for tree-like Lovász-Schrijver systems that refute unsatisfiable CNFs. More generally, we prove that an $n^{\Omega(1)}$ lower bound for the $(k+1)$-party NOF communication complexity of set-disjointness implies a $2^{n^{\Omega(1)}}$ size lower bound for all tree-like proof systems whose formulas are degree k polynomial inequalities.

1 Introduction

Linear programming, the problem of optimizing a linear objective function over the points of a given polyhedron, was shown to be polynomial-time solvable over the rationals by Khachian [15]. When integrality constraints are added, however, the resulting integer linear programming problem becomes NP-hard. Many algorithms for such problems attempt to apply efficiencies from rational linear programming to the integral case.

One of the most powerful of such approaches is to begin with the polytope defined by the original linear program without integrality constraints and systematically pare down the polytope by repeatedly refining the linear program with "cutting planes" that remove only nonintegral solutions until we are left with the convex hull of the integral solutions. These are local methods in which the initial polytope Q (expressed by the natural cutting planes constraints) is transformed through a sequence of local operations to smaller and smaller polytopes (each contained in the original one), until the integral hull of Q is reached. (At this point,

[*] Supported by NSF grants CCR-0098066 and ITR-0219468

[†] Supported by an Ontario Premiere's Research Excellence Award, an NSERC grant, and the Institute for Advanced Study. Research done while at the Institute for Advanced Study.

[‡] Supported by NSF Postdoctoral Fellowship DMS-0303258. Research done while at the Institute for Advanced Study.

ISSN 1433-8092

rational linear programming will find the correct solution.) For decision problems, this sequence terminates with the empty polytope if and only if the initial polytope contains no integral points.

One such method is that of Gomory-Chvátal cuts [6] which derives each new cutting plane as a linear combination and shift of existing facet constraints. There are even more subtle methods available, particularly in the case of 01-programming, which is also NP-complete. In a seminal paper, Lovász and Schrijver [16] introduced a variety of cutting planes methods that derive new cutting planes by first "lifting" the inequalities to higher degree polynomial inequalities (in particular quadratic inequalities) and then "projecting" them down to linear inequalities using polynomial identities and the fact that $x^2 = x$ for $x \in \{0,1\}$. These systems are now known as *Lovász-Schrijver systems (LS)*.

It may be too costly to apply these techniques to pare all the way down to the integral hull. However, even applying a smaller number of rounds of the procedure can often lead to a smaller polytope that has good approximability ratio, one for which the best nonintegral solution is not too far away from the best integral solution, so that by rounding we can achieve a good approximation to the optimal value.

There are two complexity measures that are commonly studied for Lovász-Schrijver and related cutting planes proof systems: *size* and *rank*. Intuitively, rank is the number of intermediate polytopes that must be passed through before arriving at the integral hull. In [16] it was shown that for any (relaxed) polytope P, if the rank of P is d, then the optimization and decision problems for P can be solved exactly deterministically in time $n^{O(d)}$. This very nice algorithmic property of Lovász-Schrijver systems makes them especially appealing for solving or approximating NP-hard optimization problems via linear programming. A variety of rank lower bounds for exact solution are known, even for the case of unsatisfiable systems [4, 8, 11, 7, 12]. Moreover, interesting bounds on the ranks required for good approximations to vertex cover [1] and MaxSAT [5] have been obtained. This, in turn, implies inapproximability results for these problems for *any* polynomial-time algorithm based on rank.

While there is a rich and growing body of results concerning rank, very little is known about the size of LS proofs. Informally, the size of a LS procedure with respect to some polytope P is the smallest number of hyperplanes defining all of the polytopes that we need to pass through before arriving at the integral hull. Clearly size lower bounds imply rank lower bounds, and even tree-size lower bounds imply rank lower bounds, but the converse is not known to be true. The one unconditional (tree-like) size lower bound known for LS [12] is for a family of polytopes for which decision and optimization are trivial and for which the integral hull has a trivial derivation in Chvátal's cutting planes proof system.

Problems in which the facets represent clauses of a CNF formula and a decision algorithm for 01-programming yields a propositional proof system are particularly important to analyze. Proving (tree-like) size lower bounds for such polytopes was given as one of the main open problems in [12]. The only LS size lower bounds known at present for such polytopes formulas are conditional results. First, it is an easy observation that $\mathsf{NP} \neq \mathsf{coNP}$ implies superpolynomial LS size lower bounds for some family of unsatisfiable CNF formulas. It has also been shown by [19, 9, 10] that these lower bounds also hold under other natural complexity assumptions.

In this paper we develop a new method for attacking size lower bounds for LS and for systems that generalize LS. Our main result is a proof that lower bounds on the 3-party communication complexity of set disjointness (in the number-on-forehead model) imply lower bounds on the size of tree-like LS proofs for a particular family of unsatisfiable CNF formulas. We also generalize this result to a much more powerful family of proof systems known as semantic LS^k, where lines are now degree k polynomial inequalities. All versions of LS are special cases of LS^2, and Chvátal's Cutting Planes proof system is a special case of LS^1.

More generally, we show that proving lower bounds on the $(k+1)$-party communication complexity of set disjointness implies lower bounds on the size of tree-like semantic LS^k proofs. By a natural extension of the ideas in [2] one can show that the $(k+1)$-party set disjointness problem is "complete" for the $(k+1)$-party communication complexity class $(k+1)$-NP^{cc} and a lower bound showing that it is not in $(k+1)$-RP^{cc} would already given excellent lower bounds for LS^k proofs. Such a result is already known in the case $k=1$ [2] (and was used in [13] to derive tree-like size lower bounds for Chvátal's Cutting Planes system) and set disjointness is one of the most well-studied problems in communication complexity.

Our proof can be seen as a generalization of [13] to arbitrary k but the extension requires a number of new ideas and a substantially more complicated argument that includes a detailed analysis of large sets of vertex-disjoint paths in expander graphs.

2 Definitions

2.1 Multiparty Communication Complexity and Set Disjointness

The *k-party number-on-the-forehead (NOF) model of communication complexity* computes functions (or relations) of input vectors $(x_1,\ldots,x_k) \in X_1 \times \ldots \times X_k$ distributed among k parties, such that party $i \in [k]$ sees all x_j for all $j \in [k]$, $j \neq i$.

The *k-party set disjointness problem* $\textsc{Disj}_{k,n} : (\{0,1\}^m)^k \to \{0,1\}$ is defined by $\textsc{Disj}_{k,n}(\vec{x}) = 1$ iff there is some $j \in [n]$ such that $x_{i,j} = 1$ for all $i \in [k]$. (We follow standard terminology although it might be more appropriate to call this set intersection rather than disjointness.)

A $(0,\varepsilon)$*-error k-party NOF communication protocol* for set disjointness is a protocol that for every disjoint input produces output 0 and for intersecting inputs outputs 1 with probability at least $1-\varepsilon$.

It is conjectured that for any $k \geq 2$ the k-party set disjointness problem requires nearly linear randomized NOF communication complexity. This conjecture is equivalent showing that nondeterministic k-party communication complexity can be almost optimally separated from randomized k-party communication complexity. The conjecture is proven for $k=2$ [14], but the best known lower bound for $k \geq 3$ is $\Omega(\log n)$ for general models and $\Omega(n^{1/k})$ for more restricted models [3].

2.2 Threshold Logics

The two most prevalent classes of threshold logics are Gomory-Chvátal cutting planes [6], and matrix cuts, defined by Lovász and Schriver [16]. First we briefly describe Gomory-Chvátal cutting planes, which is referred to in the literature as simply Cutting Planes (CP). A CP proof of unsatisfiability of a set of integer linear inequalities $f = \{f_1 \geq a_1, \ldots f_m \geq a_m\}$ is a sequence of integer linear inequalities $g_1 \geq b_1, \ldots, g_q \geq b_q$ such that each $g_i \geq b_i$ is either an inequality from f, an axiom ($x \geq 0$ or $1-x \geq 0$), or is obtained by one of the two rules: (i) $g_i \geq b_i$ is a positive integer linear combination of some previously derived inequalities; or (ii) $g_i \geq b_i$ is obtained from a previous inequality $cg_i \geq b_i$ by rounding (to obtain $g_i \geq \lceil b_i/c \rceil$).

There are several cutting planes proof systems defined by Lovász and Schrijver [16], collectively referred to as matrix cuts. These systems allow one to "lift" the linear inequalities to degree-two polynomials and then project back to degree one, using the fact that $x^2 = x$ for $x \in \{0,1\}$. To see that the definitions below are equivalent to the original definitions of Lovász and Schrijver, see [9].

Definition 2.1. *Given a polytope $P \subseteq Q_n$ defined by $a_i x \geq b_i$ for $i = 1,2,\ldots,m$:*

(1) An inequality $d - c^T x \geq 0$ is called an N-cut for P if

$$d - c^T x \;=\; \sum_{i,j} \alpha_{ij}(b_i - a_i^T x)x_j + \sum_{ij} \beta_{ij}(b_i - a_i^T x)(1-x_j) + \sum_j \lambda_j (x_j^2 - x_j),$$

where $\alpha_{ij}, \beta_{ij} \geq 0$ and $\lambda_j \in R$ for $i = 1, \ldots, m$, $j = 1, \ldots, n$.

(2) *A weakening of N-cuts, called N_0-cuts can be obtained if when simplifying to the linear term $d - c^T x$, we view $x_i x_j$ as distinct from $x_j x_i$.*

(3) *An inequality $d - c^T x$ is called an N_+-cut if*

$$d - c^T x \quad = \quad \sum_{i,j} \alpha_{ij}(b_i - a_i^T x)x_j + \sum_{ij} \beta_{ij}(b_i - a_i^T x)(1 - x_j) + \sum_j \lambda_j (x_j^2 - x_j) + \sum_k (g_k + h_k^T x)^2,$$

where again $\alpha_{ij}, \beta_{ij} \geq 0$, $\lambda_j \in R$ for $i = 1, \ldots, m$, $j = 1, \ldots, n$ and $g_k + h_k^T x$ is a linear function for $k = 1, \ldots, n+1$.

The operators N, N_0 and N_+ are called the *commutative, non-commutative* and *semidefinite* operators, respectively. All three are collectively called *matrix-cut* operators.

Definition 2.2. *A Lovász-Schrijver (LS) refutation for f is a sequence of inequalities $g_1, \ldots, g_q$ such that each g_i is either an inequality from f or follows from previous inequalities by an N-cut as defined above, and such that the final inequality is $0 \geq 1$. Similarly, a LS$_0$ refutation uses N_0-cuts and LS$_+$ uses N_+-cuts.*

Definition 2.3. *Let $\mathcal{P}$ be one of the proof systems CP, LS, LS$_0$ or LS$_+$. Let S be an $\mathcal{P}$-refutation of f, viewed as a directed acyclic graph. If the underlying directed acyclic graph is a tree, then S is a tree-like $\mathcal{P}$-refutation of f. The inequalities in S are represented with all coefficients in binary notation. The size of S is the sum of the sizes of all inequalities in S; the rank of S is the depth of the underlying directed acyclic graph. For a set of boolean inequalities f, the $\mathcal{P}$-size of f is the minimal size over all $\mathcal{P}$ refutations of f. Similarly the $\mathcal{P}$-treesize of f is the minimal size over all tree-like $\mathcal{P}$-refutations of f.*

Note that in our definition of these cutting planes systems, we can derive a new inequality from any number of previous inequalities in one step, whereas in a typical proof system, we are restricted to fanin-two. However, in light of Caratheodory's theorem, we can assume without loss of generality that the fanin is at most $n+1$ in CP and n^2+n+1 in LS, and so the size and treesize would not increase significantly if instead our proof systems would be defined to have fanin 2.

All of above proof systems, CP, LS, LS$_0$, and LS$_+$, are special cases of more general *semantic* threshold logic proof systems which we will define now.

A *k-threshold formula* over Boolean variables $x_1, \ldots, x_n$ is a formula of the form $\sum_j \gamma_j m_j \geq t$, where γ_j, t are integers, and for all j, m_j is a multilinear monomial of degree at most k. The *size* of a k-threshold formula is the sum of the sizes of γ_j and t, written in binary notation.

Let f_1, f_2, g be k-threshold formulas in the variables $\vec{x}$. We say that *g is semantically entailed* by f_1 and f_2 if for every $0/1$ assignment to $\vec{x}$ that satisfies both f_1 and f_2, g is also satisfied.

Let f be an unsatisfiable CNF formula over $x_1, \ldots, x_n$, and let $t_1, \ldots, t_m$ be the underlying set of clauses of f, written as 1-threshold inequalities. A **Th(k)** *refutation of f, $\mathcal{P}$,* is a sequence of k-threshold formulas, $L_1, \ldots, L_q$, where each L_j is one of the inequalities t_i, $i \in [m]$, or is semantically entailed by two formulas L_i and $L_{i'}$ with $i, i' < j$, and the final formula L_q is $0 \geq 1$. The *size* of $\mathcal{P}$ is the sum of the sizes of all k-threshold formulas occurring in $\mathcal{P}$. The proof is *tree-like* if the underlying directed acyclic graph, representing the implication structure of the proof, is a tree. (That is, every formula in the proof, except for the formulas from f, is used at most once as an antecedent of an implication.)

CP refutations are a special case of **Th(1)** semantic refutations, and thus lower bounds for tree-like **Th(1)** semantic refutations imply similar lower bounds for tree-like CP. (This was already shown in [13].)

As mentioned earlier, since we can assume that any of the Lovász-Schrijver systems can be assumed to have fan-in two, it follows that any of the systems LS_0, LS and LS^+ can easily be converted into **Th(2)** semantic refutations with at most a polynomial increase in size, and if the original proof is tree-like, so is the semantic refutation. Thus, lower bounds for tree-like **Th(2)** semantic refutations imply similar lower bounds for all tree-like Lovász-Schrijver systems.

2.3 Relating the Complexity of Threshold Logics to the Complexity of a Search Problem

Let f be an unsatisfiable CNF formula. We will be interested in the following search problem, $Search_f$ associated with f: given a truth assignment α, find a clause from f which is falsified by α. The model for this computation is a decision tree whose nodes evaluate polynomial threshold functions:

A *k-threshold decision tree* is a rooted, directed tree whose vertices are labeled with k-threshold functions and edges are labeled with either 0 or 1. The leaves of the tree are labeled with clauses of f. A k-threshold decision tree solves $Search_f$ in the obvious way: start at the root and evaluate the threshold function; follow the edge that is consistent with the value of the threshold function; continue until the computation reaches a leaf and output the associated clause. The size S of a k-threshold decision tree is the sum of the sizes of all threshold formulas in the tree, where the coefficients are written in binary. The depth of a k-threshold decision tree is the depth of the underlying tree.

The following lemma, similar to the degree 1 case in [13], shows that from a small tree-like **Th(k)**-semantic refutation of an unsatisfiable formula f, a small-size, small-depth k-threshold decision tree for $Search_f$ can be extracted.

Lemma 2.1. *Let $\mathcal{P}$ be a tree-like* **Th(k)***-semantic refutation of f of size S. Then there is a k-threshold decision tree for $Search_f$ of depth $O(\log S)$ and size $O(S)$.*

Proof. Assume that $\mathcal{P}$ is a size S tree-like **Th(k)**-semantic refutation of f. We will describe a depth $O(\log S)$, size $O(S)$, k-threshold decision tree which computes the search problem associated with f. The proof is by induction on S; clearly if $S = 1$ then the unsatisfiable formula is a single, false threshold formula, so the lemma holds. For the inductive statement, assume that the size of $\mathcal{P}$ is $S > 1$. By the 1/3-2/3 trick, there is an intermediate threshold formula f in $\mathcal{P}$ such that the number of formulas above f is between $S/3$ and $2S/3$. Let the subtree of $\mathcal{P}$ with root formula f be denoted by $\mathcal{A}$ and let the remainder of $\mathcal{P}$ (consisting of all formulas of $\mathcal{P}$ that are not in $\mathcal{A}$, and with f replaced by $1 \geq 1$) be denoted by $\mathcal{B}$. In our decision tree, we first query f. If f evaluates to 0, we proceed on the subtree $\mathcal{A}$ and otherwise we proceed on the subtree $\mathcal{B}$. By induction, both $\mathcal{A}$ and $\mathcal{B}$ have size at most $2S/3$, so the height of the decision tree obtained will be $\log_{3/2}(S)+1 \leq O(\log S)$. To see that the decision tree computes the search function, notice that if f evaluates to false on a given truth assignment ϕ, then we proceed on the subproof $\mathcal{A}$. By soundness of the proof, at least one of the leaf formulas of $\mathcal{A}$ must be falsified by ϕ. A similar argument holds when f evaluates to true. □

The next lemma, adapted from arguments in [18], shows that any relation computed by a shallow k-threshold decision tree can also be efficiently computed by a $k+1$ player communication complexity protocol (number-on-forehead model), over any partition of the variables. Some details are given in the appendix.

Lemma 2.2. *Suppose that relation $R(x_1,\ldots,x_{kn})$ is computed by a a depth d k-threshold decision tree in which all coefficients are bounded by $N \geq n$. For any partition of the inputs into k sets,*
(a) there is a $k+1$-party deterministic NOF communication complexity protocol for R in which $O(d \log N)$ bits are communicated in total, and

(b) there is a $k+1$-party 0-error randomized NOF communication complexity protocol for R in which $O(d(\log\log N)^2)$ bits are communicated in total and which computes an answer with probability at least $1-\varepsilon$.

Proof Sketch. Observe that for each monomial m in each k-threshold formula there is at least one party that can evaluate the monomial. Thus each k-threshold formula can be evaluated as the sum of k values known to different parties. The k-threshold formulas can be evaluated using variants of the standard deterministic or randomized 2-party communication algorithms for the GREATERTHAN function. □

The following is a corollary.

Theorem 2.3. *Suppose that f has a tree-like* **Th(k)***-semantic refutation of size S. Then there exists a $k+1$-party 0-error randomized NOF communication complexity protocol for Search$_f$ (over any partition of the variables into k groups) that communicates $O(\log^3 S)$ bits and produces an answer with probability at least $1-1/n$.*

Further, if all k-threshold formulas in the **Th(k)***-semantic refutation have coefficients bounded by a polynomial in n, then the 0-error randomized communication complexity can reduced to $O(\log S(\log\log n)^2)$ or the protocol can be made deterministic using $O(\log S \log n)$ bits.*

2.4 k-fold Tseitin formulas

Our hard examples will be based on the well-known Tseitin graph formulas. Let $G=(V,E)$ be any connected, undirected graph and let $\vec{c}\in\{0,1\}^V$. The *Tseitin formula for G with respect to charge vector $\vec{c}$*, $TS(G,\vec{c})$, has variables $\mathrm{Vars}(G)=\{y_e \mid e\in E\}$. The formula states that for every vertex $v\in V$, the parity of the edges incident with v is equal to the charge, c_v, at node v. It is expressed propositionally as the conjunction of the clauses obtained by expanding $\oplus_{e\ni v} y_e = c_v$ for each $v\in V$. Note that for a graph with maximum degree d, each clause is of width $\leq d$ and the number of clauses is $\leq |V|2^d$.

Notice that $TS(G,\vec{c})$ is satisfiable if and only if $\sum_{v\in V} c_v$ is even. For odd $\vec{c}$, *Search*$_{TS(G,\vec{c})}$ takes a 0/1 assignment α to $\mathrm{Vars}(G)$ and outputs a clause of $TS(G,\vec{c})$ that is violated. In particular, a solution to *Search*$_{TS(G,\vec{c})}$ will produce a vertex v such that the parity equation associated with vertex v is violated by α.

To make the search problem hard for k-party NOF communication protocols (and thus, by Theorem 2.3, hard for $k-1$-threshold decision trees) we modify $TS(G,\vec{c})$ by replacing each variable y_e by the conjunction of k variables, $\bigwedge_{i=1}^{k} y_e^i$, and expanding the result into clauses. We call the resulting *k-fold Tseitin formula*, $TS^k(G,\vec{c})$, and its variable set, $\mathrm{Vars}^k(G)=\{y_e^i \mid e\in E, i\in[k]\}$.

For a fixed graph G and different odd-charge vectors $\vec{c}\in\{0,1\}^{V(G)}$, the various problems *Search*$_{TS^k(G,\vec{c})}$ are very closely related. Define ODDCHARGE$^k(G)$ to be the k-party NOF communication search problem which takes as input an odd charge vector $\vec{c}\in\{0,1\}^{V(G)}$, seen by all players, and an assignment α to $\mathrm{Vars}^k(G)$, in which player i sees all values but the assignment α_e^i to y_e^i for $e\in E(G)$, and requires that the players output a vertex v that is a solution to *Search*$_{TS^k(G,\vec{c})}$.

3 Reduction from Set Disjointness to ODDCHARGE

We give a sequence of reductions to show that for a suitably chosen graph G, an efficient k-party NOF communication complexity protocol for ODDCHARGE$^k(G)$ will imply an efficient 1-sided error randomized k-party NOF protocol for the set disjointness relation.

We apply the Valiant-Vazirani argument to show that, without loss of generality, it suffices to derive a 1-sided error protocol for a version of set disjointness in which the input has intersection size 0 or size 1, and the job of the players is to distinguish between these two cases. We call this promise problem *zero/one set disjointness*.

Fix G to be an appropriately chosen fixed-degree graph on n vertices with good expansion and girth properties, where $m = n^{1/3}/\log n$. Our reduction from zero/one set disjointness to $\text{ODDCHARGE}^k(G)$ goes via an intermediate problem, $\text{EVENCHARGE}^k(G)$, which is the exact analog of $\text{ODDCHARGE}^k(G)$ except that the input charge vector $\vec{c}$ is even rather than odd and the requirement is *either* to find a charge violation or to determine that no charge violation exists.

The reduction from $\text{EVENCHARGE}^k(G)$ to $\text{ODDCHARGE}^k(G)$, which is similar in spirit to a reduction of Raz and Wigderson [20], works by planting a single randomly chosen additional charge violation. This yields a protocol for $\text{EVENCHARGE}^k(G)$ that works well on average for each class of inputs with a given number of charge violations.

The most difficult part of our argument is the reduction from zero/one set disjointness to $\text{EVENCHARGE}^k(G)$ for suitable graphs G. The key idea is that for even $\vec{c}$, charge violations of $TS^k(G,\vec{c})$ come in pairs: Given an instance $\vec{x} \in (\{0,1\}^m)^k$ of zero/one set disjointness, using the public coins, the players randomly choose an even charge vector $\vec{c}$ and m vertex-disjoint paths in G, $p_1,\ldots,p_m$, for each $j \in [m]$, the players plant the $x_{1,j},\ldots,x_{k,j}$ as the assignment along each edge of path p_j, in a random solution that otherwise meets the chosen charge constraint. By construction, a charge violation can occur only at the endpoints of a path and only if there is an intersection in the set disjointness problem.

By far the most difficult part is ensuring that the resulting problem looks sufficiently like a random instance of $\text{EVENCHARGE}^k(G)$ with either 0 or 2 charge violations so that we can apply the average case properties of the protocol for $\text{EVENCHARGE}^k(G)$. This places major constraints on the graph G. The bulk of the work is in showing that a small number of specific properties: rapid mixing, modest degree, and high girth – properties all met by a family of expanders constructed in [17] – are sufficient.

Distributions on labeled graphs For the rest of the paper in the Tseitin tautologies we will use a family of graphs H_n that is the union of two edge-disjoint graphs on the same set of n vertices $[n]$, G_n and T_n. G_n will be a Δ-regular expander graph of the form defined by Lubotzky, Phillips, and Sarnak [17] for $\Delta = \Theta(\log n)$. Since $\overline{G_n}$ has degree $> n/2$, there is a spanning tree T_n of maximum degree 2 (a Hamiltonian path) in $\overline{G_n}$. Clearly H_n also has maximum degree $\Theta(\log n)$ and thus $TS^k(H_n,\vec{c})$ has size $n^{O(k)}$.

Let H_n be such a graph and let $\vec{c}$ be an even charge vector. We define $Sol(H_n,\vec{c})$ to be the set of all 0/1 assignments to the edges of H_n so that for each vertex $v \in [n]$, the parity of edges incident with v is equal to c_v. A uniform random distribution over $Sol(H_n,\vec{c})$ can be obtained by first selecting 0/1 values uniformly at random for all edges in G_n and then choosing the unique assignment to the edges of T_n that fulfill the charge constraints given by $\vec{c}$.

Given a bit value b associated with an edge $e \in G_n$, we can define a uniform distribution $\mathcal{L}_k(b)$ over the corresponding variables y_e^i, $i \in [k]$. Such an assignment is chosen randomly from $\mathcal{L}_k$ on input b by the following experiment. If $b = 1$ then set all variables associated with edge e, y_e^i, $i \in [k]$ to 1. Otherwise if $b = 0$, set the vector $(\vec{y}_e)_{i\in[k]}$ by choosing uniformly at random from the set of $2^k - 1$ not-all-1 vectors.

Definition 3.1. *For any $t \geq 0$ let $\mathcal{D}_t$ be a distribution given by the following experiment on input $H_n = G_n \cup T_n$.*

1. *Choose an even charge vector $\vec{c} \in \{0,1\}^n$ uniformly at random.*

2. *Choose some $\beta \in Sol(H_n,\vec{c})$ uniformly at random.*

3. *For each $e \in G_n$, select the values for the vector $(y_e)_{i\in[k]}$ from $\mathcal{L}_k(\beta_e)$ and for each $e \in T_n$, set $y_e^i = \beta_e$ for all $i \in [k]$.*

4. *Select a random subset $U \subseteq [n]$ of $2t$ vertices and produce charge vector $\vec{c}^{\,U}$ from $\vec{c}$ by toggling all bits c_v for $v \in U$.*

5. *Return the pair $(\alpha, \vec{c}^{\,U})$ where α is the boolean assignment to the variables y_e^i, $i \in [k]$, $e \in H_n$.*

Reduction from EVENCHARGE **to** ODDCHARGE

Lemma 3.1. *Let G be any connected graph on n vertices and let $\Delta(G)$ be the maximum degree in G. Suppose that Π_{odd} is a randomized k-party NOF protocol for* ODDCHARGE$^k(G)$ *that produces an answer with probability at least $1-\varepsilon$, is correct whenever it produces an answer, and uses at most s bits of communication. Then there is a randomized k-party NOF protocol Π_{even} for* EVENCHARGE$^k(G)$ *that uses $s+\Delta(G)$ bits of communication and has the following performance:*

$$\begin{array}{rcl}\Pr\limits_{(\alpha,\vec{c})\in\mathcal{D}_0}[\Pi_{even}(\alpha,\vec{c}) = \text{true}] & = & 1 \\ \Pr\limits_{(\alpha,\vec{c})\in\mathcal{D}_t}[\Pi_{even}(\alpha,\vec{c}) \in Err(\alpha,\vec{c})] & \geq & 2/3-\varepsilon \text{ for } t \geq 1.\end{array}$$

Proof. Let Π_{odd} be a protocol for ODDCHARGE$^k(G)$ and assume that $V(G) = [n]$. We give a protocol Π_{even} for EVENCHARGE$^k(G)$. On input $(\alpha, \vec{c})$ and random public string r: Using r, choose a random vertex $v \in [n]$. Check whether the parity equation associated with vertex v is satisfied by α using at most $\Delta(G)$ bits of communication. If it is not, return v. Otherwise, create an odd charge vector, $\vec{c}^{\,\{v\}}$, which is just like $\vec{c}$ except that the value of c_v is toggled. Now run Π_{odd} on input $(\vec{c}^{\,\{v\}}, \alpha)$. If Π_{odd} returns the planted error v or if Π_{odd} does not return a value then return "true"; if Π_{odd} returns $u \neq v$, output u.

Suppose that $(\alpha, \vec{c}) \in \mathcal{D}_0$. Then α satisfies all charges specified by $\vec{c}$, so when Π_{odd} returns a vertex the above protocol must output "true" because Π_{odd} has one-sided error–that is, Π_{odd} will only return a vertex u when there is an error on the parity equation associated with u. Now suppose that $(\alpha, \vec{c}) \in \mathcal{D}_t$ so exactly $2t$ parity equations are violated. If the random vertex v does not satisfy its parity constraints, then the algorithm is correct. The remaining case is when v satisfies the parity equation and in this case we call Π_{odd} on a pair $(\alpha, \vec{c}^{\,\{v\}})$ where exactly $2t+1$ parity equations are violated.

We show the probability bound separately for each $T \in [n]^{(2t+1)}$. Because the events $\text{Err}(\alpha, \vec{c'}) = T$ partition the probability space, this proves the claim. By symmetry, for $T \in [n]^{(2t+1)}$ and *any* function g with codomain T, we have that $\Pr_{\alpha,\vec{c},v}[g(\alpha,\vec{c}^{\,\{v\}}) = v \mid \text{Err}(\alpha,\vec{c}^{\,\{v\}}) = T] = 1/(2t+1)$ since it is equally likely for $\vec{c'} = \vec{c}^{\,\{v\}}$ to be generated as $\vec{c}^{\,\{u\}}$ for any $u \in T$. Thus we obtain:

$$\begin{array}{rl}\Pr\limits_{\alpha,\vec{c},v}[\Pi_{even}(\alpha,\vec{c}^{\,\{v\}}) \text{ errs} \mid \text{Err}(\alpha,\vec{c}^{\,\{v\}}) = T] & \\ = & \Pr\limits_{\alpha,\vec{c},v}[\Pi_{odd}(\alpha,\vec{c}^{\,\{v\}}) = v \text{ or } \Pi_{odd}(\alpha,\vec{c}^{\,\{v\}}) \text{ is not defined } \mid \text{Err}(\alpha,\vec{c}^{\,\{v\}}) = T] \\ \leq & 1/(2t+1)+\varepsilon \;\leq\; 1/3+\varepsilon\end{array}$$

for $t \geq 1$. □

Reduction from Zero/One Set Disjointness to EVENCHARGE**:** We now show how to use a k-party NOF communication complexity protocol Π_{even} for EVENCHARGE$^k(H_n)$ as guaranteed by Lemma 3.1 to produce a k-party NOF protocol for the zero/one set disjointness problem which uses the following definition.

Definition 3.2. *Let $P_l^{(m)}$ be the set of all sequences of m vertex-disjoint length l paths in G_n.*

Lemma 3.2. *Let $m = n^{1/3}/\log n$. For sufficiently large n and for any even charge vector $\vec{c}$, if there is a probabilistic k-party NOF communication complexity protocol, Π_{even} for* EVENCHARGE$^k(H_n)$ *using s bits, satisfying the conditions in Lemma 3.1 for $\mathcal{D}_0$ and $\mathcal{D}_1$, then there is a randomized $(0, 1/3+\varepsilon+o(1))$ error k-party NOF communication complexity protocol Π_{01disj} for zero/one set disjointness on input $\vec{x} \in (\{0,1\}^m)^k$ that uses s bits of communication.*

Proof. Let $\vec{x}$ be an instance of zero/one set disjointness. Protocol Π_{01disj} will call Π_{even} on the graph H_n, on a pair $(\alpha, \vec{c})$ chosen according to the following distribution/experiment:

1. On input $\vec{x}$ with public coins r:
   (a) Using public coins r, choose a random even charge vector $\vec{c} \in \{0,1\}^n$.
   (b) Using public coins r, choose a sequence of m vertex-disjoint length l paths, $p_1, \dots p_m$ uniformly at random from $P_l^{(m)}$.
   (c) Using the public coins r, choose $\beta \in Sol(H_n - \bigcup_{j=1}^m p_j, \vec{c})$
2. For all edges $e \in H_n$, all players other than player i compute α_e^i as follows:
   (a) If $e \in p_j$ for $j \in [m]$, set $\alpha_e^i = x_{i,j}$
   (b) If $e \in G_n$ and $e \notin \bigcup_{j=1}^m p_j$, choose the vector $\alpha_e^1 \dots \alpha_e^k$ according to the distribution $\mathcal{L}_k(\beta_e)$.
   (c) For the remaining edges $e \in T_n$, set all variables α_e^i for $i \in [k]$ equal to β_e.
3. Return $(\alpha, \vec{c})$

We write $\mathcal{R}(\vec{x})$ to denote the distribution on assignment/charge pairs produced by reduction Π_{01disj} when given an input $\vec{x}$. The following lemma shows that for $t = |\cap\vec{x}| \in \{0,1\}$, although $\mathcal{R}(\vec{x})$ is not the same as $\mathcal{D}_t$, $\mathcal{R}(\vec{x})$ is close to the distribution $\mathcal{D}_t$ in the ℓ_1 norm. This lemma is the main technical lemma in the proof. The proof of this lemma can be found in the next section.

Lemma 3.3. *Let $\vec{x} \in (\{0,1\}^m)^k$ and $|\cap\vec{x}| = 1$. Then $||\mathcal{R}(\vec{x}) - \mathcal{D}_1||_1$ is $o(1)$.*

Protocol Π_{01disj} will output 0 if Π_{even} returns "true" and 1 otherwise. If $\cap\vec{x} = \emptyset$, by the above construction, the support of $\mathcal{R}(\vec{x})$ is contained in that of $\mathcal{D}_0$ and thus on $\mathcal{R}(\vec{x})$, Π_{even} must answer "true" and the vector $\vec{x}$ is correctly identified as being disjoint. In the case that $\cap\vec{x}$ contains exactly one element, $\Pr[\Pi\ 01disj(\vec{x})) = 0] \geq 2/3 - \varepsilon - o(1)$. This completes the proof of Lemma 3.2. □

Reduction from Set disjointness to Zero/One Set disjointness

Lemma 3.4. *If there is an $(0, \varepsilon)$ randomized NOF protocol for the k-party zero-one-promise set-disjointness problem that uses s bits of communication where ε is a constant < 1, then there is a $(0, \frac{1}{3})$ randomized NOF protocol for the k-party set-disjointness problem that uses $O(s \log n)$ bits of communication.*

Naturally, our starting point is the well-known result of Valiant and Vazirani [21].

Lemma 3.5 (Valiant-Vazirani). *Let a be a positive integer. Fix a nonempty $S \subseteq \{0,1\}^a$, and choose $w_1, \dots w_a \in \{0,1\}^a$ independently and uniformly. With probability at least $1/4$, there exists $j \in \{0, \dots, a\}$ so that $|\{x \in S \mid \forall i \leq j,\ x \cdot w_i = 0\}| = 1$.*

Proof of Lemma 3.4. Let Π be the protocol for the promise problem. Set $a = \lceil \log n \rceil$. Using public coins, independently and uniformly choose $w_1, \dots w_l \in \{0,1\}^a$. For $j \in \{0, \dots a\}$, the players run the protocol Π, using the following rule for evaluating the input $x_{i,r}$ for $i \in [k], r \in [m]$: interpret r as a vector in $\{0,1\}^a$, and replace the value of $x_{i,r}$ by zero if for some $j' \leq j$, $w_{j'} \cdot r \neq 0$, and use the value $x_{i,r}$ if for all $j' \leq j$, $w_{j'} \cdot r = 0$. If the protocol Π returns 1, the players halt and output 1, otherwise, the players proceed to round $j+1$. If no intersection is found after all $a+1$ rounds, the players announce that the inputs are disjoint.

Clearly, this protocol uses $O(s \log n)$ bits of communication, and by the 0-error property of Π on disjoint inputs, it never outputs 1 when the inputs are disjoint. When the inputs are non-disjoint, the Valiant-Vazirani construction ensures that with probability at least $1/4$, at some round j the protocol Π is used on an input with a unique intersection, and therefore, conditioned on this event, the correct answer is returned with probability at least $1 - \varepsilon$. Therefore, the correct answer is returned with probability at least $\frac{1}{4} - \frac{\varepsilon}{4}$. Because ε is bounded away from 1 and the error is one-sided, a constant number of repetitions decreases the probability of error to $1/3$. □

Combining the reductions

Theorem 3.6. *Let $k \geq 2$ and $m = n^{1/3}/\log n$. For each n there is an odd charge vector $\vec{c} \in \{0,1\}^n$ such that for any $\varepsilon < 1/2$ the size of any tree-like* **Th(k-1)** *refutation of $TS^k(H_n, \vec{c})$ is at least $2^{\Omega((R^k_\varepsilon(\text{DISJ}_{k,m})/\log n)^{1/3})}$. Further if the coefficients in the* **Th(k-1)** *refutations are bounded by a polynomial in n then the refutation size must be at least $2^{\Omega(R^k_\varepsilon(\text{DISJ}_{k,m})/(\log n(\log\log n)^2))}$ or at least $2^{\Omega(D^k_\varepsilon(\text{DISJ}_{k,m})/\log^2 n)}$.*

Proof. By Theorem 2.3 and the definition of $\text{ODDCHARGE}^k(H_n)$, if for every $\vec{c} \in \{0,1\}^n$ there is tree-like **Th(k-1)** refutation of $TS^k(H_n, \vec{c})$ of size at most S, then there is a $1/n$-error randomized k-party NOF communication complexity protocol for $\text{ODDCHARGE}^k(H_n)$ in which at most $O(\log^3 S)$ bits are communicated. By communicating the value of one edge at the vertex to be output by this $\text{ODDCHARGE}^k(H_n)$ protocol, the players can check that this vertex is indeed in error and not produce an answer otherwise. This will produce a 0-error protocol that outputs the correct answer with probability at least $1 - 1/n$. By Lemma 3.1 this yields a randomized 0-error k-party NOF protocol Π_{even} for $\text{EVENCHARGE}^k(H_n)$ that uses $O(\log^3 S + \log n)$ bits, produces the correct answer for all inputs in the support of $\mathcal{D}_0$ and for inputs randomly chosen according to $\mathcal{D}_1$ produces a correct answer with probability at least $2/3 - 1/n$. Applying Lemma 3.2 this yields a $(0, 1/3 + 1/n + o(1))$-error k-party protocol for zero/one set disjointness on $(\{0,1\}^m)^k$ also of complexity $O(\log^3 S + \log n)$. Finally applying Lemma 3.4 yields an error $1/3$ randomized k-party NOF protocol for $\text{DISJ}_{k,m}$ of complexity $O(\log^3 S \log n + \log^2 n)$ bits in total. Applying a similar reduction using the other parts of Theorem 2.3 yields the claimed result. □

We can in fact prove something slightly stronger:

Theorem 3.7. *The same lower bounds as Theorem 3.6 hold for* every *odd charge vector $\vec{c} \in \{0,1\}^n$.*

Proof. Observe that distributions $\mathcal{D}_t$ and $R(\vec{x})$ on the assignments to $\text{Vars}^k(H_n)$ both have the property that for each edge e of T_n, $\alpha^1_e = \cdots = \alpha^k_e$. Therefore in the proof of Theorem 3.6 observe that we can replace $TS^k(H_n, \vec{c})$ by $\widetilde{TS}^k(H_n, \vec{c}) = TS^k(H_n, \vec{c}) \wedge EQ(T_n)$ where $EQ(T_n)$ is the conjunction of $(\neg y^i_e \vee y^j_e)$ for every $i \neq j \in [k]$ and every $e \in T_n$. The size of any **Th(k-1)** refutation of $TS^k(H_n, \vec{c})$ is at least that of $\widetilde{TS}^k(H_n, \vec{c})$. Moreover, it is not hard to see that for any odd weight vectors $\vec{c}, \vec{d} \in \{0,1\}^n$, $\widetilde{TS}^k(H_n, \vec{c})$ and $\widetilde{TS}^k(H_n, \vec{d})$ have proof sizes that differ by at most a polynomial additive term: Given a small proof of $\widetilde{TS}^k(H_n, \vec{d})$, let $S \subset [n]$ be the set of vertices v for which $c_v \neq d_v$. Since both c and d are odd weight vectors, $|S|$ is even. Let

$M \subset E(T_n)$ be the set of edges of corresponding to $|S|/2$ disjoint sub-paths in T_n that match the elements in S. Applying the substitution of $y^i_e = \neg y^i_e$ for each $e \in M$ and $i \in [k]$ will convert a refutation of $\widetilde{TS}^k(H_n, \vec{d})$ into a refutation of $\widetilde{TS}^k(H_n, \vec{c})$ (and vice versa). The size of the proof is unchanged. (We needed the fact that all edges along T_n had the same assignment for each y^i_e so that the value of $\wedge_{i=1}^k y^i_e$ would be complemented when we complemented the value of each y^i_e.) □

4 Proximity of distributions $\mathcal{D}_1$ and $\mathcal{R}(\vec{x})$ when $|\cap \vec{x}| = 1$

In this section we prove Lemma 3.3 that for $|\cap \vec{x}| = 1$ the distributions $\mathcal{R}(\vec{x})$ and $\mathcal{D}_1$ are close in the ℓ_1 norm. Let $\mu_{\mathcal{D}_1}$ and $\mu_{\mathcal{R}(\vec{x})}$ be their associated probability measures. We will show that for all but a set of $(\alpha, \vec{c})$ with $\mu_{\mathcal{D}_1}$ measure $o(1)$, $\mu_{\mathcal{D}_1}(\alpha, \vec{c}) = (1 \pm o(1))\mu_{\mathcal{R}(\vec{x})}(\alpha, \vec{c})$.

Given an instance of the set disjointness variables, $\vec{x} = (\{0,1\}^m)^k$, for $j \in [m]$ we say that the *color* of j is the tuple $(x_{1,j}, \ldots, x_{k,j}) \in \{0,1\}^k$. By construction, the assignment $\mathcal{R}(\vec{x})$ has color $(x_{1,j}, \ldots, x_{k,j})$ on each edge of the path p_j.

Definition 4.1. *Given an ordered sequence of paths $\vec{p} \in P_l^{(m)}$, an $\vec{x} \in (\{0,1\}^m)^k$, and an assignment α, write $\chi(\alpha_{\vec{p}}) = \vec{x}$ if and only if every edge on path p_j has color $(x_{1,j}, \ldots, x_{k,j})$ for every $j \in [m]$.*

We first observe that for any $(\alpha, \vec{c})$ with $|\mathrm{Err}(\alpha, \vec{c})| = 2$ the probability $\mu_{\mathcal{D}_1}(\alpha, \vec{c})$ depends only on the number of edges $e \in G_n$ having color 1^k in α.

Definition 4.2. *Let $\phi(a,b) = 2^{-a}(2^k - 1)^{-(a-b)}$.*

Lemma 4.1. *For any $(\alpha, \vec{c})$ with $|Err(\alpha, \vec{c})| = 2$ and $m_1 = |\{e \in E(G_n) \mid \alpha_e = 1^k\}|$,*

$$\mu_{\mathcal{D}_1}(\alpha, \vec{c}) = \frac{\phi(|E(G_n)|, m_1)}{2^{n-1}\binom{n}{2}}.$$

Proof. Let $U = \mathrm{Err}(\alpha, \vec{c})$. The probability under $\mathcal{D}_1$ that U is chosen to be flipped is $1/\binom{n}{2}$ and, given U, all of the 2^{n-1} even charge vectors $\vec{c}^U$ are equally likely. Conditioned on these events, the chance that α labels the edges for the randomly selected element of $Sol(H_n, \vec{c})$ is $2^{-|E(G_n)|}(2^k - 1)^{-(|E(G_n)|-m_1)}$. □

Definition 4.3. *For $U \subset V$ with $|U| = 2$ let $P_l^{(m)}(U)$ be the set of all elements of $P_l^{(m)}$ that have a path whose endpoints are U.*

Now consider the measure $\mu_{\mathcal{R}(\vec{x})}(\alpha, \vec{c})$. Let $\{i\} = \cap \vec{x} \subseteq [n]$, $U = \mathrm{Err}(\alpha, \vec{c})$ with $|U| = 2$, and $m_1 = |\{e \in E(G_n) \mid \alpha_e = 1^k\}|$. By the definition of $\mathcal{R}(\vec{x})$,

$$\begin{aligned}
\mu_{\mathcal{R}(\vec{x})}(\alpha, \vec{c}) &= \Pr_{\vec{p} \in P_l^{(m)}}[\mathrm{Ends}(p_i) = \mathrm{Err}(\alpha, \vec{c}) \wedge \chi(\alpha_{\vec{p}}) = \vec{x}] \\
&\quad \times \Pr_{\vec{c}' \in \{0,1\}^n,\, \alpha' \in \mathcal{L}_k(Sol(H_n - \vec{p}, \vec{c}'))}[\alpha' = \alpha_{G_n - \vec{p}} \text{ and } \vec{c}' = \vec{c}] \\
&= \Pr_{\vec{p} \in P_l^{(m)}}[\mathrm{Ends}(p_i) = \mathrm{Err}(U)] \times \Pr_{\vec{p} \in P_l^{(m)}(U)}[\chi(\alpha_{\vec{p}}) = \vec{x}] \\
&\quad \times \phi(|E(G_n)| - ml, m_1 - l)/2^{n-1}.
\end{aligned}$$

Observe that p_i is a uniformly chosen element of P_l and we can analyze the first term using the following property of random paths on LPS expanders proved as part of Lemma 4.9 in Section 4.2.2.

Lemma 4.2. *For $u \neq v \in V(G_n)$ and $l \geq c_1 \log n / \log\log n$,*
$\Pr_{p \in P_l}[Ends(p) = \{u,v\}] = (1 \pm o(1))/\binom{n}{2}$.

Thus

$$\begin{aligned} \mu_{\mathcal{R}(\vec{x})}(\alpha,\vec{c}) &= (1 \pm o(1)) \frac{\phi(|E(G_n)| - ml, m_1 - l)}{\binom{n}{2} 2^{n-1}} \cdot \Pr_{\vec{p} \in P_l^{(m)}(U)}[\chi(\alpha_{\vec{p}}) = \vec{x}] \\ &= (1 \pm o(1)) \frac{\mu_{\mathcal{D}_1}(\alpha,\vec{c})}{\phi(ml,l)} \cdot \Pr_{\vec{p} \in P_l^{(m)}(U)}[\chi(\alpha_{\vec{p}}) = \vec{x}]. \end{aligned}$$

It follows that we will obtain the desired result if we can show that for all but a $o(1)$ measure of $(\alpha,\vec{c})$ under $\mu_{\mathcal{D}_1}$,

$$\Pr_{\vec{p} \in P_l^{(m)}(U)}[\chi(\alpha_{\vec{p}}) = \vec{x}] = (1 \pm o(1))\phi(ml,l) = (1 \pm o(1)) 2^{-ml}(2^k - 1)^{-(m-1)l}$$

where $U = \mathrm{Err}(\alpha,\vec{c})$. In the case that this happens, we say that $(\alpha,\vec{c})$ is *well-distributed for $\vec{x}$.*

Using the second moment method we prove the following lemma which shows that for all but a $o(1)$ measure of $(\alpha,\vec{c})$ under $\mu_{\mathcal{D}_1}$, $(\alpha,\vec{c})$ is indeed well-distributed for $\vec{x}$. The detailed proof is given in Section 4.1.

Lemma 4.3. *Let $m \leq n^{1/3}/\log n$ and $l = 2\lceil c_1 \log n / \log\log n \rceil$ and $\vec{x} \in (\{0,1\}^m)^k$ with $|\cap \vec{x}| = 1$. For almost all $U \subset [n]$ with $|U| = 2$,*

$$\Pr_{(\alpha,\vec{c}) \in \mathcal{D}_1}[(\alpha,\vec{c}) \text{ is well-distributed for } \vec{x} \mid Err(\alpha,\vec{c}) = U] = 1 - o(1)$$

Lemma 3.3 follows from this almost immediately.

Proof of Lemma 3.3. Let $\vec{x} \in (\{0,1\}^m)^k$ and $|\cap \vec{x}| = 1$. By Lemma 4.3 and the preceding argument, for all but a set B of U that forms $o(1)$ fraction of all subsets $[n]$ of size 2,

$$\Pr_{(\alpha,\vec{c}) \in \mathcal{D}_1}[\mu_{\mathcal{R}(\vec{x})}(\alpha,\vec{c}) = (1 \pm o(1))\mu_{\mathcal{D}_1}(\alpha,\vec{c}) \mid \mathrm{Err}(\alpha,\vec{c}) = U] = 1 - o(1).$$

By Lemma 4.2, $\Pr_{(\alpha,\vec{c}) \in \mathcal{D}_1}[\mathrm{Err}(\alpha,\vec{c}) \in B] = o(1)$. Therefore by summing over distinct choices of U, we obtain that with probability $1 - o(1)$ over $(\alpha,\vec{c}) \in \mathcal{D}_1$, $\mu_{\mathcal{R}(\vec{x})}(\alpha,\vec{c}) = (1 \pm o(1))\mu_{\mathcal{D}_1}(\alpha,\vec{c})$. This is equivalent to the desired conclusion that $||\mathcal{D}_1 - \mathcal{R}(\vec{x})||_1$ is $o(1)$. □

4.1 Most $(\alpha,\vec{c})$ are well-distributed

In this section we use the second moment method to prove Lemma 4.3. For this purpose we will need the following property of the LPS expander graphs G_n, proved in Section 4.2 which will allow us to show that the correlations considered in the second moment method are low.

Definition 4.4. *For $\vec{p},\vec{q} \in P_l^{(m)}$ we write $\vec{p} \sim_s \vec{q}$ when $\vec{p}$ and $\vec{q}$ share exactly s edges. Let $\gamma > 0$ be a positive real number. We say that $U \subset V(G_n)$ is* γ-nice *if for all $s \geq 0$, $\Pr_{\vec{p},\vec{q} \in P_l^{(m)}(U)}[\vec{p} \sim_s \vec{q}] \leq \gamma^s$.*

Theorem 4.4. *(proved in § 4.2) Suppose that $m \leq n^{1/3}/\log n$ and $l = 2\lceil c_1 \log n / \log\log n \rceil$. There are constants $c > 0$ and c' such that for all but a $o(1)$ fraction of sets $U = \{u,v\} \subset V(G_n)$, for all $\vec{q} \in P_l^{(m)}(U)$ and every integer $s \geq 0$,*
$\Pr_{\vec{p} \in P_l^{(m)}(U)}[\vec{p} \sim_s \vec{q}] \leq (c'/(\log\log n)^{1/4} + (\log n)^{-c})^s$,
i.e. almost every $U \in V^{(2)}$ is $(c'/(\log\log n)^{1/4} + 1/\log^c n)$-nice.

We now use this in our application of the second moment method to prove that most $(\alpha,\vec{c})$ pairs are well-distributed:

Lemma 4.5. *Let $m \le n^{1/3}/\log n$ and $l = 2\lceil c_1 \log n/\log\log n\rceil$, $\vec{x} \in (\{0,1\}^m)^k$ with $|\cap\vec{x}| = 1$, and $|U| = 2$. If U is γ-nice with $\gamma = o(2^{-k})$, then*

$$\Pr_{(\alpha,\vec{c})\in\mathcal{D}_1}[(\alpha,\vec{c}) \textit{ is well-distributed for } \vec{x} \mid Err(\alpha,\vec{c}) = U] = 1 - o(1)$$

Proof. For each $\vec{p} \in P_l^{(m)}(U)$, let $X_{\vec{p}}$ denote the indicator variable for the event that $\chi(\alpha_{\vec{p}}) = \vec{x}$.

We now calculate $E_{(\alpha,\vec{c})\in\mathcal{D}_1}[X_{\vec{p}}]$. For $(\alpha,\vec{c})$ chosen according to $\mathcal{D}_1$, the assignment $\alpha_{\vec{p}}$ is distributed according to $(\mathcal{L}_k)^{ml}$; therefore, since for $\chi(\alpha_{\vec{p}})$ to equal $\vec{x}$, $\alpha_{\vec{p}}$ must have precisely l edges whose color is 1^k and $l(m-1)$ edges whose color is a lift of label 0,

$$E_{(\alpha,\vec{c})\in\mathcal{D}_1}[X_{\vec{p}}] = \Pr_{(\alpha,\vec{c})\in\mathcal{D}_1}[X_{\vec{p}} = 1] = \phi(ml,l) = 2^{-ml}(2^k-1)^{-(m-1)l}.$$

Let $X = \sum_{\vec{p}\in P_l^{(m)}(U)} X_{\vec{p}}$. X is the random variable denoting the number of sequences $\vec{p} \in P_l^{(m)}(U)$ for which $\chi(\alpha_{\vec{p}}) = \vec{x}$. By the linearity of expectation, $E_{(\alpha,\vec{c})}[X] = \phi(ml,l)\cdot|P_l^{(m)}(U)|$.

We use the second moment method to show that X is concentrated near its expectation. For $\vec{p},\vec{q} \in P_l^{(m)}(U)$, the random variables $X_{\vec{p}}$ and $X_{\vec{q}}$ are correlated if and only if $\vec{p}$ and $\vec{q}$ share an edge. Because U is γ-nice $\Pr_{\vec{p},\vec{q}\in P_l^{(m)}(U)}[\vec{p} \sim_s \vec{q}] \le \gamma^s$.

When $X_{\vec{p}} = 1$, the colors of all edges of $\vec{p}$ are determined. Therefore given $X_{\vec{p}} = 1$, if $\vec{p} \sim \vec{q}$, either some edge that $\vec{p}$ and $\vec{q}$ share ensures that $X_{\vec{q}} = 0$, or the probability that $X_{\vec{p}} = X_{\vec{q}} = 1$ is non-zero. In the latter case consider $G' = \bigcup_{i=1}^m (p_i \cup q_i)$ which contains $2ml - s$ edges. Since α is distributed as $\mathcal{L}_k^{G'}$ on the edges of G', the probability that $\chi(\alpha_{\vec{p}}) = \chi(\alpha_{\vec{q}}) = \vec{x}$ is larger than $[\phi(ml,l)]^2$ by a factor of either 2 or $2(2^k-1)$ per shared edge depending on whether that edge has label 1 or 0.

Let $D = \sum_{\vec{p}\sim\vec{q}} \Pr_{(\alpha,\vec{c})}[X_{\vec{p}} = X_{\vec{q}} = 1]$.

$$\begin{aligned}
D &= \sum_{s=1}^{ml}\sum_{\vec{p}\sim_s\vec{q}} \Pr_{(\alpha,\vec{c})}[X_{\vec{p}} = X_{\vec{q}} = 1] \\
&\le \sum_{s=1}^{ml}\sum_{\vec{p}\sim_s\vec{q}} (2(2^k-1))^s \Pr_{(\alpha,\vec{c})\in\mathcal{D}_1}[X_{\vec{p}} = 1] \Pr_{(\alpha,\vec{c})\in\mathcal{D}_1}[X_{\vec{q}} = 1] \\
&= \sum_{s=1}^{ml}\sum_{\vec{p}\sim_s\vec{q}} (2(2^k-1))^s[\phi(ml,l)]^2 \\
&= \sum_{s=1}^{ml} |P_l^{(m)}(U)|^2 \Pr_{\vec{p},\vec{q}\in P_l^{(m)}(U)}[\vec{p}\sim_s\vec{q}](2(2^k-1))^s[\phi(ml,l)]^2 \\
&= [|P_l^{(m)}(U)|\phi(ml,l)]^2 \sum_{s=1}^{lm} \Pr_{\vec{p},\vec{q}\in P_l^{(m)}(U)}[\vec{p}\sim_s\vec{q}](2(2^k-1))^s \\
&= [E_{(\alpha,\vec{c})\in\mathcal{D}_1}(X)]^2 \sum_{s=1}^{ml} \Pr_{\vec{p},\vec{q}\in P_l^{(m)}(U)}[\vec{p}\sim_s\vec{q}](2(2^k-1))^s \\
&\le [E_{(\alpha,\vec{c})\in\mathcal{D}_1}(X)]^2 \sum_{s=1}^{ml} \gamma^s(2(2^k-1))^s.
\end{aligned}$$

Since $\gamma = o(2^{-k})$ by hypothesis, $\sum_{s=1}^{\infty} \gamma^s (2(2^k-1))^s$ is $o(1)$ and thus D is $o([E_{(\alpha,\vec{c})\in\mathcal{D}_1}]^2)$. Therefore, $E_{(\alpha,\vec{c})}(X^2) = D + E_{(\alpha,\vec{c})}(X) = o([E_{(\alpha,\vec{c})}(X)^2) + E_{(\alpha,\vec{c})}(X)$ and by the second moment method,

$$\Pr_{(\alpha,\vec{c})\in\mathcal{D}_1}[|X - E_{(\alpha,\vec{c})\in\mathcal{D}_1}(X)| \geq \varepsilon E_{(\alpha,\vec{c})\in\mathcal{D}_1}(X)] \leq \frac{D + E_{(\alpha,\vec{c})\in\mathcal{D}_1}(X)}{\varepsilon^2 E_{(\alpha,\vec{c})}(X)^2} = o(1).$$

By choosing ε as an appropriate function that is $o(1)$, we obtain that with probability $1-o(1)$ in the choice of $(\alpha,\vec{c}) \in \mathcal{D}_1$, $X = (1 \pm o(1))\phi(ml,l) \cdot |P_l^{(m)}(U)|$ and therefore with probability $1-o(1)$ in $(\alpha,\vec{c})$, $\Pr_{\vec{p}\in P_l^{(m)}(U)}[\chi(\alpha_{\vec{p}}) = \vec{x}] = (1 \pm o(1))\phi(ml,l)$ and thus $(\alpha,\vec{c})$ is well-distributed for $\vec{x}$. □

Proof of Lemma 4.3. Let $\vec{x} \in (\{0,1\}^m)^k$ and $|\cap\vec{x}| = 1$. By Theorem 4.4 there is a $\delta > 0$ so that for all but a $o(1)$ fraction of sets $U \subset V(G_n)$ with $|U| = 2$, U is γ-nice for $\gamma = c''/(\log\log n)^{1/4}$ for some constant c'' and γ is $o(2^{-k})$. Therefore, $\Pr_{(\alpha,\vec{c})\in\mathcal{D}_1}[\text{Err}(\alpha,\vec{c})$ is γ-nice$] = 1-o(1)$ and by Lemma 4.5,
$\Pr_{(\alpha,\vec{c})\in\mathcal{D}_1}[(\alpha,\vec{c})$ is well-distributed for $\vec{x} \mid \text{Err}(\alpha,\vec{c}) = U] = 1-o(1)$. □

4.2 Graph Theoretic Properties of LPS Expanders

4.2.1 The Lubotzky-Phillips-Sarnak Expanders

The crucial properties of the expander graphs G_n constructed in [17] that we need are:

1. G_n is regular of degree $\Delta = \Theta(\log n)$.

2. G_n is connected and non-bipartite.

3. The second eigenvalue of G_n is $O(\sqrt{\log n})$.

4. The girth of G_n is $\Omega(\log n/\log\log n)$.

A walk in G_n is chosen by selecting a start node and repeatedly following one of the Δ edges adjacent to the current node.

Proposition 4.6. *There exists $c_1 > 0$ so that for every $u, v \in V(G_n)$, a random walk in G_n of length $l \geq c_1 \log n/\log\log n$ starting at u ends at vertex v with probability at least $1/n - 1/n^2$ and at most $1/n + 1/n^2$.*

We consider random walks and random paths in the G_n graphs of a fixed length $l = l(n) = 2\lceil c_1 \log n/\log\log n\rceil$ that is twice the minimum length specified in Proposition 4.6 so that their midpoints are nearly uniformly distributed.

4.2.2 Approximating Paths by Walks

The reduction Π_{01disj} chooses a random sequence of length l *paths* in G_n; however the usual mixing property of expanders, Proposition 4.6, only discusses random *walks* of length l. We show that for $\Theta(\log n)$ degree LPS expanders, the distribution of random paths of a given length is close to that for random walks of that length (something that requires mixing time smaller than degree for example).

Remark 1. *In principle one might replace disjoint paths in the definition of Π_{01disj} by disjoint walks of the same length, conditioned on each having distinct endpoints. However, in that case it would be overwhelmingly likely that many walks will repeat edges and therefore, as graphs, they would contain different numbers of edges. This would significantly complicate the second moment argument of Lemma 4.5.*

More precisely we will show that, because G_n is expanding and has high girth, random walks in G_n not only mix well but they are paths almost surely as well. We state some folklore properties of random walks and observe how they translate into properties of random paths.

For $v \in V(G_n)$, let $W_l(v)$ be the set of all Δ^l walks of length l in G_n starting at v and $P_l(v)$ be the set of all paths of length l in G_n with one endpoint v. Let $\mu_{W_l(v)}$ be the measure given by a uniform distribution over $W_l(v)$ and $\mu_{P_l(v)}$ be the measure given by a uniform distribution over $P_l(v)$.

Lemma 4.7. *There exists a universal constant c_3 so that for every $v \in V(G_n)$ and for each path $p \in P_l(v)$, $(1 - c_3/\log\log n)\mu_{P_l(v)}(p) \le \mu_{W_l(v)}(p) \le \mu_{P_l(v)}(p)$. Moreover, for w uniformly chosen from $W_l(v)$ the probability that w is not a path is at most $c_3/\log\log n$.*

Proof. Observe that every $p \in P_l(v)$ has equal measure under $\mu_{W_l(v)}$ so $\mu_{W_l(v)}(p) \le \mu_{P_l(v)}(p)$ and, moreover, $\mu_{W_l(v)}(p) = \mu_{P_l(v)}(p)\mu_{W_l(v)}(P_l(v))$.

Set $g = girth(G_n)$. By the properties of G_n, $g \ge c_0 \log n/\log\log n$ for some constant $c_0 > 0$ and its degree $\Delta \ge c_2 \log n$ for some constant $c_2 \ge 0$. Notice that for any walk w of length l each vertex in w can have at most $l/(g-3)$ many neighbors also in w. (If u is a vertex in w that has two neighbors u' and u'' in G_n within distance $g-3$ on w then there is a cycle of length $g-1$ in $w \cup \{(u,u'),(u,u'')\}$ which is a subgraph of G_n.) Therefore

$$\begin{aligned}\mu_{W_l(v)}(P_l(v)) &\ge \left(\frac{\Delta - l/(g-3)}{\Delta}\right)^l \ge 1 - \frac{l^2}{\Delta(g-3)} \\ &\ge 1 - \frac{2\lceil c_1 \log n/\log\log n\rceil^2}{c_2 \log n \cdot (c_0 \log n/\log\log n - 3)} \ge 1 - c_3/\log\log n\end{aligned}$$

for some constant c_3. □

The following are folklore properties of random walks in G_n.

Proposition 4.8. *Let W_l be the set of all walks of length l in G_n.*

1. *For each $v \in V(G_n)$, $\Pr_{w \in W_l}[v \in V(w)] \le (l+1)/n$.*
2. *For each $u \ne v \in V(G_n)$, $\Pr_{w \in W_l}[Ends(w) = \{u,v\}] = (1 \pm 2/n)/\binom{n}{2}$.*

Proof. There is a sequence of $l+1$ vertices (not necessarily distinct) on each walk w in W_l and precisely Δ^l walks in which v is the i-th vertex in w. Therefore, in total there are at most $(l+1)\Delta^l$ walks with $v \in V(w)$. (This is an overcount since v may appear more than once in w.) Since there are precisely $n\Delta^l$ random walks in G_n of length l, $\Pr_{w \in W_l}[v \in V(w)] \le (l+1)/n$.

By Proposition 4.6 the chance that a particular pair of distinct vertices $\{u,v\}$ appear as endpoints of w is $\frac{2}{n}(1/n \pm 1/n^2)$ which is $(1 \pm 2/n)/\binom{n}{2}$. □

We obtain the following easy corollary which includes a proof of Lemma 4.2.

Lemma 4.9. *Let P_l be the set of all paths in G_n of length l.*

1. *Let $V' \subseteq V(G_n)$. There exists a constant c so that*

$$\Pr_{p \in P_l}[V(p) \cap V' \ne \emptyset] \le (1 + c/\log\log n)\frac{|V'|(l+1)}{n}.$$

2. *Let $u \neq v \in V(G_n)$. Then* $\Pr_{p\in P_l}[Ends(p) = \{u,v\}] = (1 \pm o(1))/\binom{n}{2}$

Proof. By Proposition 4.8, for w a randomly chosen walk of length l in G_n,

$$\Pr_{w\in W_l}[V(w) \cap V' \neq \emptyset] \leq \frac{|V'|(l+1)}{n},$$

and by Lemma 4.7, $\Pr_{w\in W_l}[w \text{ is a path}] \geq 1 - c_3/\log\log n$. The random distribution of paths p of length l in G_n is the same as the random distribution of walks w of length l in G_n conditioned on w being a path. Therefore

$$\begin{aligned}\Pr_{p\in P_l}[V \cap V(p) \neq \emptyset] &= \Pr_{w\in W_l}[V \cap V(w) \neq \emptyset \mid w \text{ is a path}] \\ &\leq \frac{|V|(l+1)}{(1-c_3/\log\log n)n} \\ &\leq (1+c/\log\log n)\frac{|V|(l+1)}{n},\end{aligned}$$

for some constant c.

For $u \neq v \in V(G_n)$, by Lemma 4.7 $\Pr_{p\in P_l}[\text{Ends}(p) = \{u,v\}]$ is within a $1 \pm o(1)$ factor of $\Pr_{w\in W_l}[\text{Ends}(w) = \{u,v\}]$ and by Proposition 4.8 the latter is $(1 \pm o(1))/\binom{n}{2}$ which yields the desired property.

□

4.2.3 The Proof of Theorem 4.4

In this subsection we prove Theorem 4.4. We will actually prove a slightly stronger result in which $\vec{q} \in P_l^{(m)}(U)$ is replaced by any subgraph of G_n with at most $m(l+1)$ vertices and maximum degree at most 2.

It will be convenient to consider sequences of length l paths P_l^m that are not necessarily vertex-disjoint. Let $\mu_{P_l^{(m)}}$ be the uniform measure on $P_l^{(m)}$ and $\mu_{P_l^m}$ be the uniform distribution on P_l^m.

Lemma 4.10. *Suppose that $m \leq n^{1/3}/\log n$ and $l = 2\lceil c_1 \log n/\log\log n\rceil$. For any $\vec{p} \in P_l^{(m)}$,* $(1-o(1))\mu_{P_l^{(m)}}(\vec{p}) \leq \mu_{P_l^m}(\vec{p}) \leq \mu_{P_l^{(m)}}(\vec{p})$.

Proof. Conditioned on the paths in $\vec{p} \in P_l^m$ being vertex-disjoint $\mu_{P_l^m}$ is uniform over $P_l^{(m)}$. By Lemma 4.9, the probability that the i-th path shares a vertex with paths $p_1,\ldots,p_{i-1}$ is at most $(1+c/\log\log n)(l+1)^2(m-1)/n \leq 2l^2m/n$ and the probability that the paths in P_l^m are not vertex-disjoint is at most $2l^2m^2/n \leq 1/n^{1/3}$. □

We first observe that if we only we required that $\vec{p} \in P_l^{(m)}$ rather $\vec{p} \in P_l^{(m)}(U)$ – i.e., we had no requirement that one path in $\vec{p}$ have its endpoints in U – then the exponentially-decaying bound on intersection size of Theorem 4.4 would be relatively easy.

Lemma 4.11. *Suppose that $m \leq n^{1/3}/\log n$ and $l = 2\lceil c_1 \log n/\log\log n\rceil$. There is some constant $c \geq 0$ such that for all subgraphs G' of G_n with at most $m(l+1)$ vertices and every integer $s \geq 0$,*

$$\Pr_{\vec{p}\in P_l^{(m)}}[|E(\cup\vec{p}) \cap E(G')| \geq s] \leq (\log n)^{-cs}.$$

Proof. For $\vec{p} \in P_l^{(m)}$, because each component of $\vec{p}$ is a path of length l, if $|E(\cup\vec{p}) \cap E(G')| \geq s$ then there are at least $\lceil s/l \rceil$ paths p_i in $\vec{p}$ that that share an edge (and therefore a vertex) with $\cup\vec{p}$. By Lemma 4.9, the probability that a random p_i from P_l shares a vertex with G' is at most $(1 + c/\log\log n)(l+1)^2 m/n < 2l^2 m/n$. Therefore for elements of P_l^m, the probability that there are least $r = \lceil s/l \rceil$ such paths is at most $\binom{m}{r}(2l^2 m/n)^r < (2l^2 m^2/n)^r/2$. By Lemma 4.10, the probability that this happens for elements of $P_l^{(m)}$ is at most $(2l^2 m^2/n)^r \leq n^{s/(3l)} = (\log n)^{-cs}$ for some constant $c > 0$. □

The major complication of the proof of Theorem 4.4 is the assumption that $\vec{p}$ contains a path with endpoints u and v for $U = \{u, v\}$, $u \neq v$. We base the analysis of paths with endpoints U on the analysis of walks with endpoints U. For some sets U, for example if u and v are adjacent in G_n, the distributions of random walks and random paths with endpoints U may not be close to each other.[1] We will see that for most choices of U, the probabilities under the two distributions are close to each other and this will be enough to obtain the bound required by Theorem 4.4.

Definition 4.5. *For $U = \{u, v\} \in V(G_n)$ let $W_l(U)$ be the set of all walks in G_n of length n that have endpoints U.*

Lemma 4.12. *There is a constant c_4 such that for all but at most a $c_4/\log\log n$ fraction of pairs $u \neq v \in V(G_n)$*

$$\Pr_{w \in W_l(\{u,v\})}[w \textit{ is a path}] \geq 2/3.$$

Proof. By Lemma 4.7,

$$\Pr_{w \in W_l}[w \text{ is not a path}] \leq c_3/\log\log n.$$

Therefore by definition,

$$\sum_{u \neq v \in V(G_n)} \Pr_{w \in W_l}[\text{Ends}(w) = \{u, v\}] \Pr_{w \in W_l(\{u,v\})}[w \text{ is not a path}] \leq c_3/\log\log n.$$

By Proposition 4.8, $\Pr_{w \in W_l}[\text{Ends}(w) = \{u, v\}] \geq (1 - 2/n)\binom{n}{2}^{-1}$ and thus

$$(1 - 2/n)\binom{n}{2}^{-1} \sum_{u \neq v \in V(G_n)} \Pr_{w \in W_l(\{u,v\})}[w \text{ is not a path}] \leq c_3/\log\log n,$$

which says that the expected value

$$E_{u \neq v \in V(G_n)}\big(\Pr_{w \in W_l(\{u,v\})}[w \text{ is not a path}]\big) \leq \frac{c_3/\log\log n}{(1 - 2/n)}.$$

We now apply Markov's inequality to obtain that the fraction of pairs $u \neq v \in V(G_n)$ for which $\Pr_{w \in W_l(\{u,v\})}[w \text{ is not a path}] \geq 1/3$, is at most $\frac{c_3/\log\log n}{(1-2/n)/3} \leq c_4/\log\log n$ for some constant c_4.

□

[1] Even in these cases the distributions may be sufficiently close but we do not need to analyze them.

Bounding Intersection Size of Random Walks

Lemma 4.12 will allow us to use the following analysis involving a random walk with endpoints in U rather than a random path. As in the proof for part (a), we also find it convenient to do the calculation assuming independent random choices of paths.

Lemma 4.13. *Let G' be a subgraph of G_n with the property that every vertex has degree at most d in G'. For fixed $v \in V(G_n)$,*

$$\Pr_{w \in W_l(v)}[|E(w) \cap E(G')| \geq s] \leq \binom{l}{s} \left(\frac{d}{\Delta}\right)^s.$$

Proof. There are at most $\binom{l}{s}$ many choices of steps in the random walk in which the first s shared edges can occur. Fix some such set of steps $S \subseteq [l]$. For each $i \in S$ a necessary condition for the i-th edge in the walk to lie in $E(G')$ is that the endpoint u after step $i-1$ must lie in $V(G')$. Since $\deg_{G'}(u) \leq d$, given that $u \in V(G')$, the probability that the i-th edge lies in $E(G')$ is then at most d/Δ. That is, conditioned on a shared edge in each of the first j elements in S, the chance of a shared edge in the $j+1$-st element in S is at most d/Δ because every vertex has degree at most d in G'. This yields a total probability at most $\binom{l}{s}(d/\Delta)^s$ as required. □

In order to analyze the random walks in $W_l(U)$ we need more than the result of Lemma 4.13 since it constrains only one endpoint of the random walk rather than both endpoints. We can view each half of a random walk in which both endpoints are constrained as two random walks of half the length with only one endpoint constrained. (Obviously, these two half-length walks are highly correlated.)

Lemma 4.14. *Let $l = 2\lceil c_1 \log n / \log\log n \rceil$. Let G' be a subgraph of G_n in which every vertex has degree at most d. For $u \neq v \in V(G_n)$,*

$$\Pr_{w \in W_l(\{u,v\})}[|E(w) \cap E(G')| \geq s] < \left(\frac{2dl}{\Delta}\right)^{s/2}.$$

Proof. Without loss of generality, walk $w \in W_l(\{u,v\})$ starts at u and ends at v. Let $l' = l/2$. Let $w = (w_u, w_v)$ where w_u and w_v each have length l'. We first observe that w_u is nearly uniformly distributed in $W_{l'}(u)$:

Let $w^* \in W_{l'}(u)$ and let v^* be the end of w^*.

$$\begin{aligned}
&\Pr_{w \in W_l(\{u,v\})}[w_u = w^* \mid w \text{ starts at } u] \\
&= \frac{\Pr_{w \in W_l(u)}[w_u = w^* \text{ and } w_v\text{, starting at } v^*\text{, ends at } v]}{\Pr_{w \in W_l(u)}[w \text{ ends at } v]} \\
&= \frac{\Pr_{w_u \in W_{l'}(u)}[w_u = w^*] \cdot \Pr_{w_v \in W_{l'}(v^*)}[w_v \text{ ends at } v]}{\Pr_{w \in W_l(u)}[w \text{ ends at } v]}
\end{aligned}$$

Clearly $\Pr_{w_u \in W_{l'}(u)}[w_u = w^*] = \Delta^{-l'} = \Delta^{-l/2}$ and since $l > l' \geq c_1 \log n / \log\log n$ by Proposition 4.6, both $\Pr_{w_v \in W_{l'}(v^*)}[w_v \text{ ends at } v]$ and $\Pr_{w \in W_l(u)}[w \text{ ends at } v]$ are $1/n \pm 1/n^2$ and thus

$$\Pr_{w \in W_l(\{u,v\})}[w_u = w^* \mid w \text{ starts at } u] = (1 \pm O(1/n))\Delta^{-l/2}.$$

Since G_n is a *regular* undirected graph, a length l random walk from u to v has the same distribution as a length l random walk from v to u. Thus by symmetry with the above argument, within a $1 \pm O(1/n)$ factor, w_v is distributed as a (nearly) uniform random walk of length l' starting at v.

Now if there are a total of s edges in common between w and G' then at least $\lceil s/2 \rceil$ must be shared between G' and one of the two halves of w, w_u and w_v. By Lemma 4.13 and the above argument each of these probabilities is at most $(1+O(1/n))(\frac{dl'}{\Delta})^{\lceil s/2 \rceil}$ and the total probability is at most $2(1+O(1/n))(\frac{dl}{2\Delta})^{\lceil s/2 \rceil} \leq (2\frac{dl}{\Delta})^{\lceil s/2 \rceil}$. □

Deriving the bound

Lemma 4.15. *Let $l = 2\lceil c_1 \log n / \log\log n \rceil$ and $m \leq n^{1/3}/\log n$. For any fixed subgraph G' of G_n with at most $m(l+1)$ vertices and maximum degree at most 2, and any set $U = \{u,v\} \subset V(G_n)$,*

$$\Pr_{(w,\vec{p}))\in W_l(U)\times P_l^{m-1}}[|(E(w)\cup E(\vec{p}))\cap E(G')| \geq s] \leq (c''/\log\log n)^{s/4} + (\log n)^{-cs/2}.$$

Proof. If there are s edge intersections between $E(w) \cup E(\vec{p})$ and G', then at least $s/2$ of them occur in either w or $\vec{p}$. Lemma 4.14 implies that $\Pr_{w\in W_l(U)}[|E(w)\cap E(G')| \geq s/2] \leq \left(\frac{4l}{\Delta}\right)^{s/4} \leq (c''/\log\log n)^{s/4}$.

By Lemma 4.11, $\Pr_{\vec{p}\in P_l^{m-1}}[|E(\vec{p})\cap E(G')| \geq s/2] \leq \Pr_{\vec{p}\in P_l^m}[|E(\vec{p})\cap E(G')| \geq s/2] \leq (\log n)^{-cs/2}$. □

We now obtain Theorem 4.4:

Lemma 4.16. *Suppose that $m \leq n^{1/3}/\log n$ and $l = 2\lceil c_1 \log n/\log\log n \rceil$. For all but a $c_4/\log\log n$ fraction of all $U = \{u,v\}$, $u \neq v \in V(G_n)$, there are constants $c, c' > 0$ such that for all subgraphs G' of G_n with at most $m(l+1)$ vertices and maximum degree 2 and for every integer $s \geq 0$,*

$$\Pr_{\vec{p}\in P_l^{(m)}(U)}[|E(\cup\vec{p})\cap E(G')| \geq s] \leq ((c'/\log\log n)^{1/4} + (\log n)^{-c})^s.$$

Proof. By Lemma 4.12, for all but a $c_4/\log\log n$ fraction of U, $\Pr_{w\in W_l(U)}[w \text{ is a path}] \geq 2/3$. For any such U, since the distribution of $w \in W_l(U)$ conditional on w being a path is uniform over $P_l(U)$, the measure of any event on $P_l(U) \times P_l^{m-1}$ is at most $3/2$ times that on $W_l(U) \times P_l^{m-1}$. Further, by the same argument as Lemma 4.10, the probability that the paths in $\vec{p}$ chosen from $P_l(U) \times P_l^{m-1}$ are vertex disjoint is at least $1 - o(1)$ conditioned on being vertex disjoint the distribution of $\vec{p}$ is uniform over $P_l^{(m)}(U)$. Therefore the measure of any event on $P_l^{(m)}(U)$ is at most $(1+o(1))3/2 \leq 2$ times that on $W_l(U) \times P_l^{m-1}$. Applying Lemma 4.15 and adjusting constants c and c' yields the bound. □

5 Discussion

There are a couple of interesting open problems related to our work beyond the natural problem of the communication complexity of DISJ$_k$. First, does semantic LS^k have a separation oracle, as LS does? This is closely related to whether or not LS^k is automatizable and we conjecture that the answer to both questions is negative. Secondly, is it possible to extend our lower bounds to other tautologies that would imply inapproximability results for polynomial-time LS^k-based algorithms? (For example, if we could prove superpolynomial lower bounds for tree-like LS^k proofs of random 3CNF formulas, this would imply inapproximability results for LS^k-based linear programming algorithms for MaxSAT [5].)

Finally we would like to point out a connection between our main result and the complexity of disjoint NP pairs. An open question in complexity theory is whether or not all pairs of disjoint NP sets can be separated by a set in P. This is known to be false under the assumption P $\neq$ UP and also by the assumption $\mathsf{P} \neq \mathsf{NP} \cap \mathsf{coNP}$. It is an open question whether or not it is implied by $\mathsf{P} \neq \mathsf{NP}$. Let us consider the same question with respect to communication complexity rather than polynomial time: can every pair of relations with small nondeterministic k-party communication complexity be separated by a small probabilistic/deterministic protocol? In [20] the answer is shown to be unconditionally false for $k = 2$. In particular, they give a pair of disjoint properties on $3m$-vertex graphs G, a matching on $2m$ vertices of G and an independent set of $2m+1$ vertices of G, and show that this pair cannot be separated by any small probabilistic/deterministic protocol. In this paper, we have shown that for any k, the question is still false, under $k\text{-}\mathsf{RP}^{cc} \neq k\text{-}\mathsf{NP}^{cc}$.

Acknowledgements

We are indebted to Avi Wigderson for helpful discussions and insights.

References

[1] S. Arora, B. Bollobás, and L. Lovász. Proving integrality gaps without knowing the linear program. In *Proceedings 43nd Annual Symposium on Foundations of Computer Science*, pages 313–322, Vancouver, BC, November 2002. IEEE.

[2] L. Babai, P. Frankl, and J. Simon. Complexity classes in communication complexity theory. In *27th Annual Symposium on Foundations of Computer Science*, pages 337–347, Toronto, Ontario, October 1986. IEEE.

[3] P. Beame, T. Pitassi, N. Segerlind, and A. Wigderson. A direct sum theorem for corruption and the multiparty NOF communication complexity of set disjointness. In *Proceedings Twentieth Annual IEEE Conference on Computational Complexity*, San Jose, CA, June 2005.

[4] A. Bockmayr, F. Eisenbrand, M.E. Hartmann, and A.S. Schulz. On the Chvatal rank of polytopes in the 0/1 cube. *Discrete Applied Mathematics*, 98(1-2):21–27, 1999.

[5] J. Buresh-Oppenheim, N. Galesi, S. Hoory, A. Magen, and T. Pitassi. Rank bounds and integrality gaps for cutting planes procedures. In *Proceedings 44th Annual Symposium on Foundations of Computer Science*, pages 318–327, Boston, MA, October 2003. IEEE.

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

[7] V. Chvátal, W. Cook, and M. Hartmann. On cutting-plane proofs in combinatorial optimization. *Linear Algebra and its Applications*, 114/115:455–499, 1989.

[8] W. Cook, C. R. Coullard, and G. Turan. On the complexity of cutting plane proofs. *Discrete Applied Mathematics*, 18:25–38, 1987.

[9] S. Dash. *On the matrix cuts of Lovász and Schrijver and their use in Integer Programming*. PhD thesis, Department of Computer Science, Rice University, March 2001.

[10] S. Dash. An exponential lower bound on the length of some classes of branch-and-cut proofs. In W. Cook and A. S. Schulz, editors, *IPCO*, volume 2337 of *Lecture Notes in Computer Science*, pages 145–160. Springer-Verlag, 2002.

[11] F. Eisenbrand and A. S. Schulz. Bounds on the Chvatal rank of polytopes in the $0/1$-cube. *Combinatorica*, 23(2):245–261, 2003.

[12] D. Grigoriev, E. A. Hirsch, and D. V. Pasechnik. Complexity of semi-algebraic proofs. In *(STACS) 2002: 19th Annual Symposium on Theoretical Aspects of Computer Science*, volume 2285 of *Lecture Notes in Computer Science*, pages 419–430, Antibes, France, February 2002. Springer-Verlag.

[13] R. Impagliazzo, T. Pitassi, and A. Urquhart. Upper and lower bounds on tree-like cutting planes proofs. In *9th Annual IEEE Symposium on Logic in Computer Science*, pages 220–228, Paris, France, 1994.

[14] B. Kalyanasundaram and Georg Schnitger. The probabilistic communication complexity of set intersection. In *Proceedings, Structure in Complexity Theory, Second Annual Conference*, pages 41–49, Cornell University, Ithaca, NY, June 1987. IEEE.

[15] L. G. Khachian. A polynomial time algorithm for linear programming. *Doklady Akademii Nauk SSSR, n.s.*, 244(5):1093–1096, 1979. English translation in *Soviet Math. Dokl. 20*, 191–194.

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

[17] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[18] N. Nisan. The communication complexity of threshold gates. In V.S.D. Mikl'os and T. Szonyi, editors, *Combinatorics: Paul Erdös is Eighty, Volume I*, pages 301–315. Bolyai Society, 1993.

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

[20] R. Raz and A. Wigderson. Monotone circuits for matching require linear depth. *Journal of the ACM*, 39(3):736–744, July 1992.

[21] L. Valiant and V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, pages 85–93, 1986.

6 Appendix

6.1 Evaluating k-threshold Decisions Trees Using Multiparty Communication Complexity

Lemma 6.1. *If the search problem for $f(x_1,\dots,x_{kn})$ has a k-threshold decision tree of depth d, and where all k-threshold formulas have coefficients bounded by a polynomial in n, then there exists a $k+1$-party deterministic NOF communication complexity protocol for Search$_f$ (over any partition of the variables into k groups) where $O(d\log n)$ bits are communicated.*

Proof. Fix a partition of $x_1,\dots x_{kn}$. Let $\alpha_1 m_1+\ldots+\alpha_q m_m \geq t$ be the k-threshold formula queried at the root of the k-threshold decision tree for f. Then the set of monomials m_i can be partitioned into $k+1$ groups, where group i contains the monomials that can be "seen" by the i^{th} player. Each player (in turn) communicates the weighted linear combination of their monomials to the other players. After all players have spoken, each player can simply add up the total sum and see if it is greater than the target t, in order to evaluate the k-threshold formula. The $k+1$ players then continue on the half of the decision tree which agrees with the value of this formula. The protocol terminates after d rounds, and each round requires $O(\log n)$ bits of communication. □

Lemma 6.2. *If the search problem for f has a threshold decision tree of depth d and size S, then there exists a probabilistic communication complexity protocol for Search$_f$ with $\varepsilon \leq 1/n$ where $O(d\log^2 S)$ bits are communicated.*

Proof. As above, the players will proceed in d rounds, at each step evaluating the threshold formula and proceeding on the consistent subtree of half the size. Let f be the first threshold formula at the root of the decision tree. Since the entire threshold decision tree has size S, all coefficients must also be bounded by size $2^{O(S)}$. As before, partition the monomials of f into $k+1$ groups where the i^{th} player can "see" the monomials in group i. Each of the $k+1$ players computes the weighted sum of their respective monomials. Call these sums $y_1,\dots,y_{k+1}$, respectively. Note that for each $i\leq k+1$, $|y_i|\leq O(S)$. Player $k+1$ uses $y'_{k+1}=t-y_{k+1}$ and by applying Lemma 6.3, there is a probabilistic protocol allowing the players to determine whether the sum of the y_i's is at least t, where $O(\log S)^2$ bits are exchanged. After evaluating this formula f, the players then continue on the half of the decision tree which agrees with the value of f. The protocol terminates after d rounds, for a total of $O(d\log^2 S)$ bits of communication. □

Lemma 6.3. *Let $y_1,\dots,y_{k+1}$ be binary integers of length n. Then there is an $O((k\log n)^2)$-bit $(k+1)$-player Number-in-Hand probabilistic protocol for determining whether $y_1+\ldots+y_k\geq y_{k+1}$.*

Proof. Consider first the case where there are 3 players ($k+1=3$), and they are trying to determine whether $x+y\geq z$. The players will follow a divide and conquer strategy by recursively examining segments of their strings.

Let the first half of x be x_1, where $|x_1|=\lfloor n/2\rfloor$, and the right half of x be x_2, where $|x_2|=\lceil n/2\rceil$. Similarly let y_1,y_2,z_1,z_2 be the left and right halves of y and z respectively.

Player I randomly selects a prime number $p\in[1,n^3\log n]$ and sends $(p,x_1 \mod p)$ to players II and III. Then player II sends $y_1 \mod p$ to player III, where p is the same prime. Player III then computes (x_1 mod $p+y_1 \mod p$), the sum of their left halves modulo the prime, and compares it to z_1, modulo p.

The first case is when the sum is identical, ie., $(x_1 \mod p+y_1 \mod p)=z_1 \mod p$. In this case, with probability at least $1-1/n^2$, $x_1+y_1=z_1$; that is, the actual sums of their left halves is equal to the left half of z. To see this, notice that an error occurs whenever $x_1+y_1\neq z_1$, but $(x_1+y_1) \mod p=z_1 \mod p$, which happens if p divides $(x_1+y_1-z_1)$. Since $x_1+y_1-z_1$ has at most $O(n)$ prime divisors, and p is chosen from $O(n^3)$ primes, the probability of error is less than $1/n^2$. Assuming that $(x_1+y_1=z_1)$, then we know that $x+y$ is at least as large as z, as long as the sum of the lower order bits is larger, that is, as long as $x_2+y_2 \geq z_2$. This is because the sum of $x+y$ on the low order bits could induce a carry but this would just make $x+y$ larger than z. Thus we can continue by recursively checking whether or not $(x_2+y_2 \geq z_2)$.

The second case is when $(x_1 \mod p + y_1 \mod p) = z_1 - 1 \mod p$. Again, with probability at least $1-1/n^2$, this implies that $x_1+y_1 = z_1-1$. Assuming that this is the case, we know that $x+y \geq z$ if and only if $x_2+y_2 \geq z_2 + 2^{\lfloor n/2 \rfloor +1}$. This is because in order for the sum of the higher order bits to be at least as large as z_1, we require at least one carry from the low order bits.

Since we can get a carry of either 0 or 1 from the low order bits, the final case is when $(x_1 \mod p + y_1 \mod p)$ is not equal to either $z_1 \mod p$ or to $z_1 - 1 \mod p$. In this case, again with high probability we know that (x_1+y_1) is not equal to either z_1 or to z_1-1, and assuming this is the case, it follows that $x+y \geq z$ if and only if $x_1+y_1 \geq z_1$.

In any of the above cases, we can continue recursively on bit strings of half the original size. At each stage, we send $O(\log n)$ bits, and the total number of stages is $O(\log n)$ for a total of $O(\log n)^2$ bits sent. The probability of error at each stage is $O(1/n^2)$ and therefore the total error is less than $1/n$ (for sufficiently large n).

For $k+1 \geq 3$ the argument is very similar to the above argument, only now the carry from the low order bits can be anything from 0 to $k-1$, so we get k different cases, depending on whether the mod p sum of the first k players inputs, $(y_1 \mod p + \ldots + y_k \mod p)$ is equal to $y_{k+1} - i \mod p$, for $i = 0, \ldots, k-1$. □