Lower Bounds for Cutting Planes Proofs with Small Coefficients

MARIA BONET * TONIANN PITASSI † RAN RAZ ‡

Abstract

We consider small-weight Cutting Planes (CP^*) proofs; that is, Cutting Planes (CP) proofs with coefficients up to $Poly(n)$. We use the well known lower bounds for monotone complexity to prove an exponential lower bound for the length of CP^* proofs, for a family of tautologies based on the clique function. Because Resolution is a special case of small-weight CP, our method also gives a new and simpler exponential lower bound for Resolution.

We also prove the following two theorems : (1) Tree-like CP^* proofs cannot polynomially simulate non-tree-like CP^* proofs. (2) Tree-like CP^* proofs and Bounded-depth-Frege proofs cannot polynomially simulate each other.

Our proofs also work for some generalizations of the CP^* proof system. In particular, they work for CP^* with a deduction rule, and also for any proof system that allows any formula with small communication complexity, and any set of sound rules of inference.

1 Introduction

One of the most fundamental questions in propositional proof theory is: how strong is a particular proof system? In particular, one tries to give examples of tautologies with no short proofs in the system. It is believed that for any conceivable proof system there exist tautologies (of size n), with no proofs of size polynomial in n. However, proving this for every conceivable system is equivalent to proving that $NP \neq Co-NP$ [CR], which is an extremely hard task. Therefore, many researchers have concentrated on proving the existence of hard tautologies (i.e. tautologies with no polynomial size proofs), for specific natural classes of proof systems. Two of the biggest open problems in the area are to prove the existence of hard tautologies for Frege systems, and for extended Frege systems. So far, however, such lower bounds have been given only for weaker systems.

The Cutting Planes (CP) proof system, first introduced in [CCT], is a sound and complete refutation system for proving the unsatisfiability of propositional formulas in conjunctive normal form. It is based on showing that there are no integral solutions for a family of linear inequalities associated with an unsatisfiable CNF formula. The Cutting Planes technique was first introduced in [Gom] in the context of linear programming, and shown in [Chv] to be a canonical way of proving that every integral solution of a given system of linear inequalities satisfies another given inequality. In brief, a CP refutation is a sequence of linear inequalities, where the initial inequalities are those that we are trying to prove unsatisfiable, the final inequality is the inequality $0 \geq 1$, and all intermediate inequalities follow from one or two previous ones by a sound rule of inference.

Besides being a very natural proof system, CP appears to be relatively powerful. First, it is a natural generalization of Resolution. Secondly, the propositional pigeonhole principle (PHP) has a very simple polynomial-size CP proof [CCT]. This is interesting because PHP is the canonical hard tautology that has been previously used to prove lower bounds for Resolution as well as for bounded-depth Frege systems (e.g., [H], [BIKPPW]).

Since PHP has a short CP proof, CP is strictly stronger than Resolution (with respect to what can be proven by polynomial-size proofs). It was shown in [G] that any Frege system can polynomially simulate CP, and therefore CP lies between Resolution and Frege. Thus, understanding the power of CP is an important step in order to give lower bounds for Frege systems.

A restriction of CP is the system CP^*. CP^* proofs are CP proofs with the sole restriction that all intermediate inequalities are required to have coefficients bounded in size by $Poly(n)$, where n is the size of the formula to be proven. CP^* still appears to be quite powerful: Resolution is still a special case of CP^*, where the coefficients have size $O(1)$, and PHP still has small CP^* proofs. In fact, as far as we know, all the CP proofs ever considered are actually CP^* proofs!

The main result of this paper is an exponential lower bound on the size of CP^* proofs. Our family of unsatisfiable formulas are based on the clique function. To prove our lower bound, we show how to extract a small monotone circuit computing clique on many inputs, from a small CP^* proof. The lower bound for CP^* then follows using known monotone lower bounds for the clique function [Razb1], [AB]. This lower bound method can be viewed as an extension of the method in [IPU]. [1] We also show how our lower bound method can be applied

*Department of Mathematics, University of Pennsylvania, bonet@saul.cis.upenn.edu. Research supported by NSF grant CCR-9403447

†Mathematics and Computer Science, University of Pittsburgh, toni@cs.pitt.edu. Research supported by an NSF postdoctoral fellowship and by NSF Grant CCR-9457782

‡Department of Applied Math, Weizmann Institute, ranraz@wisdom.weizmann.ac.il. Part of this work was done while the author was a postdoc at Princeton University and DIMACS

[1] We have recently learned that the same methods were used independently and before us by Razborov [Razb2], to prove that certain statements are not provable in some fragments of bounded arithmetic.

to obtain exponential lower bounds for several generalizations of CP^*. In particular, our method works for any propositional proof system consisting of a sound family of inference rules, each of which takes a constant number of formulas to a single formula, and such that intermediate formulas have small communication complexity. Our method also works for a generalization of CP^* where we allow a form of the deduction rule.

Our second result is a separation between tree-like CP^* and non-tree-like CP^*. (A tree-like proof is a proof where each intermediate formula is used only once). We also obtain separations between tree-like CP^* and bounded-depth Frege systems. The family of formulas used in the latter lower bounds are based on the st-connectivity function.

This paper is organized as follows. In Section 2, we give some preliminary definitions and notation. In Section 3, we give an informal discussion of our method and the general idea of the proof. Section 4 contains the main result, the exponential lower bound for small-weight CP proofs, as well as several generalizations of this lower bound for stronger systems. In Section 5 we study a particular tautology based on the st-connectivity function, and show that it has short CP^* proofs but requires large tree-like CP^* proofs. As a corollary of this theorem combined with known results, we obtain several separation results. Finally we conclude in Section 6 with a short discussion of a general interpolation theorem that follows from this work and its connection with recent works of Razborov and Krajicek.

2 Definitions and Background

2.1 Cutting Planes

We will first describe the CP refutation system for CNF formulas. For a more complete treatment see [C, G, CCT].

CP formulas in the variables $x_1, \dots, x_n$ are inequalities of the form

$$\sum_{i=1}^{n} a_i x_i \geq A$$

where $a_1, \dots, a_n$, and A are integral constants, and $x_1, \dots, x_n$ are integral variables. We think of the formula as a linear inequality in the variables $x_1, \dots, x_n$. Notice that in this definition of a CP formula, the constant A always appears at the right hand side of the inequality, and the variables always appear in the same order. To simplify notation we will sometimes write constants on both sides of the inequality, and change the order of variables. Also, sometimes we will write $A \leq \sum_{i=1}^{n} a_i x_i$.

Given an initial family of CP formulas in the variables $x_1, \dots, x_n$, the CP system has four sound rules of inference :

1. Basic algebraic simplifications like deleting (or adding) terms of the form $0x_i$.

2. Addition of two inequalities: if $\sum_{i=1}^{n} a_i x_i \geq A$, and $\sum_{i=1}^{n} b_i x_i \geq B$ we can derive
$$\sum_{i=1}^{n} (a_i + b_i) x_i \geq (A + B)$$

3. Multiplication of an inequality by an integer: if $\sum_{i=1}^{n} a_i x_i \geq A$, and $c \in Z$, we can derive
$$\sum_{i=1}^{n} c a_i x_i \geq cA$$

4. Division of an inequality by an integer: if $\sum_{i=1}^{n} a_i x_i \geq A$, and $c \in Z$ divides each a_i, we can derive
$$\sum_{i=1}^{n} \frac{a_i}{c} x_i \geq \left\lceil \frac{A}{c} \right\rceil$$

As introduced in [CCT], the CP system can be used as a refutation system for CNF formulas : Given a CNF formula f, in the variables $x_1, \dots, x_n$, we think of the variables $x_1, \dots, x_n$ as integers that can get the values 0, or 1, where 0 represents $FALSE$, and 1 represents $TRUE$. We first translate the formula f into a family of CP formulas in the following way: a clause

$$\bigvee_{i=1}^{k} x_{j_i} \vee \bigvee_{i=1}^{m} \neg x_{l_i}$$

is translated into the CP formula

$$\sum_{i=1}^{k} x_{j_i} + \sum_{i=1}^{m} (1 - x_{l_i}) \geq 1$$

The family, $E(f)$, of CP formulas corresponding to the CNF formula, f, is the set of CP formulas we obtain by translating each clause in f, together with the inequalities $x_i \geq 0$, and $-x_i \geq -1$, for all $1 \leq i \leq n$.

A CP refutation of f (or $E(f)$) is a CP proof for the inequality $0 \geq 1$, from the initial family $E(f)$. It was proved in [CCT] that CP is a sound and complete refutation system for CNF formulas. Clearly, by looking at $\neg f$, CP is a sound and complete proof system for DNF formulas. This result can also be derived by noticing that CP is in fact a generalization of Resolution.

Definition The *length* of a CP proof is the number of formulas in it. The *size* of the proof is the number of binary symbols needed to write down the proof.

It was shown in [CCT] that any CP proof can be converted into a new one, such that the lengths of all the coefficients in the new proof, and the length of the new proof, are all polynomially bounded in the length of the original proof. Therefore without loss of generality, we can assume that the length and the size of a CP proof are polynomially equivalent, and will not distinguish between them in this paper.

How small can we assume that the coefficients in the proof are ? By [CCT], we can assume without loss of generality that all of the coefficients are smaller than $2^{Poly(l)}$, where l is the length of the proof. Can we assume that all the coefficients are even smaller ? It is still open whether coefficients smaller than $O(2^n)$, or $O(2^{n^\epsilon})$, or $Poly(n)$ (where n is the number of variables) are enough.

Definition A refutation of a formula f, in small-weight Cutting Planes (CP^*) is a CP refutation of $E(f)$ with the additional requirement that all the coefficients involved are of size $Poly(n)$, where n is the size of f.

CP^* is in fact a complete refutation system for CNF formulas, simply because it is a generalization of Resolution. It is still not known whether CP^* can polynomially simulate CP. Although we tend to believe that CP is stronger, we believe that CP proofs with large coefficients are highly non-intuitive. Therefore, in a way, CP^* captures (at least) the intuitive part of CP. We also believe that our hard tautologies for CP^* will turn out to be hard for CP as well.

To any CP proof (or CP^* proof) corresponds a standard directed acyclic graph: the nodes of the graph correspond to the formulas in the proof. An edge from a formula L' to a formula L exists if and only if L' was directly used to derive L. Clearly the in-degree of each node, L, is 0,1, or 2, as L can be one of the initial formulas, or can be derived by one or two previous formulas. If the out-degree of all of the nodes are at most 1, the graph is a *tree*, and the proof is called *tree-like*. Thus a tree-like proof is one where every formula is used at most once. (If one wants to construct a tree-like proof from a non-tree-like proof, one might have to re-derive a formula each time the formula is used).

2.2 Generalized Cutting Planes systems

In this section we define more general abstract systems that our lower bound applies to.

Let S be an arbitrary refutation system, and let f be a set of formulas that we are trying to refute. Then the *deduction rule* for S allows the prover to query an arbitrary allowable formula L, and then the proof splits into two halves, where the first half is dedicated to refuting $f \vee \{L\}$, and the second half is dedicated to refuting $f \vee \{\neg L\}$. We will now formally define CP with Deduction.

Definition Let f be a set of unsatisfiable linear inequalities. P is a refutation of f in CP **with Deduction** if and only if P consists of the following two parts:

1. The first part, A, is a set of threshold formulas arranged in a balanced binary tree, where each edge of the tree is labeled with exactly one threshold formula in A, and such that if $e1$ and $e2$ are the two edges leading out of a vertex, then the threshold formula associated with $e1$ must be the negation of the threshold formula associated with $e2$. Let the set of all simple paths from root to leaf in the tree be denoted by $\{p_1, .., p_q\}$, and let the set of associated formulas along path p_i be denoted by $Form(p_i)$.

2. The second part, B, consists of q separate CP proofs, B_1, ..., B_q, where B_i is a CP refutation of $f \cup Form(p_i)$.

Definition A proof P in CP^* with Deduction is defined as in the definition of CP with Deduction, except that now all of the formulas in P are required to have small coefficients.

We will now define a generalization of CP^* where the formulas are allowed to be more expressive.

Definition Let f be a boolean formula over underlying variables $x_1, .., x_n$. Let S_1, S_2 be a fixed partition of $\{x_1, .., x_n\}$ into two disjoint sets. The *communication complexity of f with respect to S_1 and S_2* is the standard deterministic communication complexity required to compute f, when Player I is given a truth assignment of the variables in S_1, and Player II is given a truth assignment of the variables in S_2. The *communication complexity* of f is the worst-case deterministic communication complexity of f with respect to S_1 and S_2, over all partitions S_1,S_2 of the variables. The ϵ-probabilistic communication complexity of f is defined in the same maner.

The threshold formulas of CP are a special case of formulas with small communication complexity. Any small-weight threshold function has a small deterministic communication complexity protocol for any partition of the input, and any high-weight threshold function has a small ϵ-probabilistic communication complexity protocol for any partition of the input. The following generalizations of CP^* and CP allow us to work with more general formulas that have small communication complexity (ϵ-probabilistic communication complexity), and more general sets of rules of inference.

Definition Let f be a set of boolean formulas using n distinct variables. Let P be a sequence of boolean formulas. P is a **Generalized** CP^* refutation of f if and only if P satisfies the following conditions:

1. Each formula in P has communication complexity of $O(poly(\log n))$;

2. Each formula in P is either from f, or follows from one or two previous formulas by a sound inference (that is, h follows from g_1 and g_2 by a sound inference if any truth assignment that falsifies h also falsifies either g_1 or g_2);

3. The final formula in P is unsatisfiable.

If only conditions 2 and 3 hold we will call P an **Inference** refutation of f.

Definition P is a **Generalized** CP refutation of f if and only if: every formula of P has ϵ-probabilistic communication complexity $O(poly(\log n))$, and conditions 2 and 3 above also hold.

In Section 4, we will prove lower bounds for both of the above generalizations of Cutting Planes. We also note here that our lower bounds in Section 4 also hold in the more general setting where the inferences take up to some fixed constant number of formulas to a single formula (in the above definition, we have fixed the constant to be 2).

3 Methods and Results

In this paper, we will use the well-known lower bounds for monotone complexity [Razb1, AB] to prove lower bounds for the length of CP^* proofs. Our result is inspired by the result of [IPU], who prove an exponential lower bound for the length of tree-like CP proofs, for some tautology. Below we give the main ideas of their proof.

Given a monotone boolean function, f, a minterm x of f, and a maxterm y of f, there must be at least one coordinate i, such that $x_i = 1$, and $y_i = 0$. This simple fact inspired [KW] to define the following communication search problem: Assume that A is a subset of minterms of a monotone boolean function f, and B is a subset of maxterms of the same function f. Player I gets a minterm $x \in A$, Player II gets a maxterm $y \in B$, and their goal is to find a coordinate i, with $x_i = 1$ and

$y_i = 0$ (the sets A and B are known to both players). Karchmer and Wigderson [KW] proved that if A is the set of all the minterms of f, and B is the set of all the maxterms of f, then the communication complexity of the corresponding search problem is exactly equal to the monotone circuit depth of the function f.

Inspired by [KW]'s result, [IPU] constructed tautologies $T(A, B)$ for particular sets $A, B \in \{0,1\}^n$, that express the following:

$$x \in A,\ y \in B \to \bigvee_{i=1}^{n} (x_i = 1\ \wedge\ y_i = 0).$$

By simple reductions, [IPU] showed how in some cases: a tree-like CP^* (CP) proof for $T(A, B)$ can be cheaply translated into a deterministic (probabilistic) communication complexity protocol for the corresponding communication search problem. These connections enabled them to use known lower bounds for communication search problems to prove lower bounds on the length of tree-like CP (or CP^*) proofs.

In particular, [IPU] took f to be the monotone boolean function that interprets the inputs as an undirected graph of $n = 3k$ vertices, and outputs 1 if and only if the graph contains a matching of size k. They took the sets A, B of minterms and maxterms that were previously considered in [RW2].

In this paper we will also use this method with minor modifications to give a tree-like CP^* lower bound for an st-connectivity tautology. This result combined with a new upper bound will give us a separation between CP^* and tree-like CP^*.

Can the same be done for non-tree-like proofs? *The main result of this paper generalizes the result of [IPU] to the non-tree-like case.* We show directly how to translate any CP^* proof for $T(A, B)$ into a monotone boolean circuit that separates the sets A and B.

In this paper, we will consider the monotone boolean function f that interprets the inputs as a non-directed graph of $n = k^{1.5}$ vertices, and outputs 1 if and only if the graph contains a clique of size k. We take the sets A, B of minterms and maxterms of f that were previously considered in [Razb1, AB]. We form the tautology $T(A, B)$ as before. Then using our main result together with known lower bounds for the monotone complexity of the clique function [Razb1, AB], we obtain exponential lower bounds for the size of any CP^* proof for $T(A, B)$. We remark that the proof actually works for coefficients up to $O(2^{n^\epsilon})$. We also remark that we use the clique-based tautology rather than the tautology used in [IPU] only because the monotone circuit lower bounds are stronger for the clique function.

4 Lower bounds for CP^*

In this section, we are going to use the lower bound for the monotone complexity of the clique function [Razb1, AB] to obtain a lower bound for a certain clique tautology in small-weights Cutting Planes.

A graph G_x on n vertices is called a k-clique if G_x consists of a single clique of size k, and no other edges. The graph G_x is said to be a minterm of the clique function because G_x contains a k-clique, but if any edge is taken away, this condition is violated. A graph G_y on n vertices is called a $(k-1)$-coclique if the vertices are partitioned into $k-1$ sets, and no edges are present within each set, but all edges are present between the sets. The graph G_y is a maxterm of the clique function because G_y does not contain a k-clique, but if any edge is added, then there will be a k-clique.

The lower bound in [Razb1, AB] is very strong. It says that for some k, every monotone circuit separating k-cliques from $(k-1)$-cocliques requires exponential size.

Definition A monotone boolean function $Q_{n,k}$ is called a **clique separator** if it interprets the inputs as the edges of a graph on n vertices, and outputs 1 on an input representing a k-clique, and 0 on an input representing a $(k-1)$-coclique.

Theorem 1 *[Razb1, AB] For $k = n^{2/3}$, any monotone boolean circuit that computes a clique separator function $Q_{n,k}$ requires size $\Omega(2^{(n/\log n)^{1/3}})$.*

Informally, our version of the clique principle states that if G_x is a k-clique, and if G_y is a $(k-1)$-coclique then there must be an edge present in G_x that is absent in G_y. We will formalize the negation of the k-clique principle on graphs with n vertices by the propositional formula $\neg CLIQUE_{n,k}$. The underlying variables are $x = \{x_{i,j} \mid 1 \le i \le k,\ 1 \le j \le n\}$, and $y = \{y_{i,j} \mid 1 \le i \le k-1,\ 1 \le j \le n\}$. The matrix x describes the graph G_x in the following way: the variable $x_{i,j}$ is 1 if and only if j is the i^{th} element of the k-clique, and 0 otherwise. Similarly, y describes the graph G_y, where $y_{i,j}$ is 1 iff vertex j is in set i, and 0 otherwise.

Let us note here that every clique and coclique have several different matrix representations. We will often use the phrase "in some matrix representation".

The unsatisfiable formula $\neg CLIQUE_{n,k}$ is the conjunction of the following clauses. The clauses in (1)-(3) describe the condition that x must be a matrix that describes a k-clique. The clauses in (4)-(5) say that y must be a matrix that describes a $(k-1)$-coclique. The clauses in (6) say that if there is an edge from vertex i to vertex j in G_x, then i and j cannot be in the same group in G_y.

1. $x_{l,1} \vee \cdots \vee x_{l,n}$ for all l, $1 \le l \le k$
2. $\neg x_{l,i} \vee \neg x_{l,j}$ for all i, j, l such that $1 \le l \le k$ and $1 \le i, j \le n$, $i \ne j$
3. $\neg x_{l,i} \vee \neg x_{l',i}$ for all i, l, l' such that $1 \le l, l' \le k$ and $1 \le i \le n$, $l \ne l'$
4. $y_{1,i} \vee \cdots \vee y_{k-1,i}$ for all i such that $1 \le i \le n$
5. $\neg y_{l,i} \vee \neg y_{l',i}$ for all i, l, l' such that $1 \le l, l' \le (k-1)$ and $1 \le i \le n$, $l \ne l'$
6. $\neg x_{l,i} \vee \neg x_{l',j} \vee \neg y_{t,i} \vee \neg y_{t,j}$ for all i, j, l, l', t such that $1 \le l, l' \le k$, $1 \le t \le (k-1)$ and $1 \le i, j \le n$, $l \ne l'$ and $i \ne j$.

To prove our CP^* lower bound, we are going to assume we have a polynomial size refutation of $\neg CLIQUE_{n,k}$. From the existence of such a refutation we will extract a monotone circuit of polynomial size computing a function $Q_{n,k}$, as above. By the previous theorem this is a contradiction, and we have to conclude that there cannot be a polynomial size refutation of $\neg CLIQUE_{n,k}$.

In the next theorem we will show how to extract monotone circuits from refutations. Let us note here that while the representation of graphs in the tautology uses a matrix encoding, the input variables, $e_{i,j}$, of the circuit represent (in the standard way) possible edges in the graphs. So we will use different representations of graphs depending on whether we are in the proof context or in the circuit context. Also, we will use capital letters to refer to 0-1 truth assignments, and lower case letters to refer to propositional variables and input variables to circuits.

4.1 Bounds for Standard CP^*

Theorem 2 *Given any Cutting Planes refutation of $\neg CLIQUE_{n,k}$, we can build a monotone circuit of size $O(m \cdot s^6)$, for some clique separator function $Q_{n,k}$. Here m is the length of the refutation, and s is the maximum absolute value that $\sum a_{i,j}X_{i,j}$ and $\sum b_{i,j}Y_{i,j}$ can take throughout the refutation (the maximum is taken over all the formulas $\sum a_{i,j}x_{i,j} + \sum b_{i,j}y_{i,j} \geq c$ of the refutation, and over all the 0-1 truth assignments for x, y).*

Proof We will give here a more general proof that disregards the actual rules of inference, and only assumes that they are all sound. We remark that for the actual rules of inference of CP, one can prove that the circuit is of size $O(m \cdot s^4)$.

We build the circuit by levels. Each line (formula) in the refutation gives rise to a different level of circuits. At the level corresponding to line L, the circuit only distinguishes pairs of cliques and cocliques that in some matrix representation falsify the line L. The last line is $0 \geq 1$. Since every pair of clique and coclique in matrix representation falsifies it, the circuit at that level will compute the clique function on all k-cliques and all $(k-1)$-cocliques.

For any line

$$L : \sum a_{i,j}x_{i,j} + \sum b_{i,j}y_{i,j} \geq c$$

in the refutation, and any pair of integers (M, N) such that $M+N < c$, we will build a monotone circuit $C^L_{M,N}$. To build the circuits $C^L_{M,N}$, we will use circuits $C^{L'}_{M',N'}$ from previous levels. Let $(\vec{V}, \vec{W})$ be a pair of truth assignments for the input variables $e_{i,j}$ (of the circuits that we are building), such that $\vec{V}$ represents a k-clique, and $\vec{W}$ represents a $(k-1)$-coclique. The circuits $C^L_{M,N}$ will satisfy the following :

1. If some matrix representation $(\vec{X}, \vec{Y})$ of $(\vec{V}, \vec{W})$ satisfies $\sum a_{i,j}X_{i,j} = M$, and $\sum b_{i,j}Y_{i,j} = N$ then $C^L_{M,N}$ on input $\vec{V}$ gives output 1 and $C^L_{M,N}$ on input $\vec{W}$ gives output 0.

   It will be simpler to disregard circuits $C^L_{M,N}$, for M, N that cannot be achieved as the sums $\sum a_{i,j}X_{i,j} = M$ and $\sum b_{i,j}Y_{i,j} = N$.

2. The extra work that is needed to build all the circuits $C^L_{M,N}$ (for all M, N), from all the circuits at previous levels is $O(s^6)$.

Clearly, for the last line, $0 \leq 1$, the circuit $C^L_{0,0}$ will compute the clique function on all k-cliques and all $(k-1)$-cocliques, and the circuit size will be at most $O(m \cdot s^6)$.

Assume that L is the l-th formula in the refutation. We will build the circuits $C^L_{M,N}$ by induction on l. Suppose that for every line, L', numbered $< l$ we have the circuits $C^{L'}_{M',N'}$. We will now build the circuits for the l-th line.

Let L be $\sum a_{i,j}x_{i,j} + \sum b_{i,j}y_{i,j} \geq c$, and fix M, N such that $M + N < c$. We are going to divide the proof into cases, depending if L is an axiom, or was derived from previous formulas.

Case 0: L is an axiom of the types $1-6$.
(This is the base case, and must occur for $l = 1$). If L is of the types $1-5$, all pairs of clique and coclique $(\vec{V}, \vec{W})$, in any matrix representation, satisfy the line L, and therefore the circuits $C^L_{M,N}$ are trivial.

If L is of type 6, L is $1-x_{l,i}+1-x_{l',j}+1-y_{t,i}+1-y_{t,j} \geq 1$, or equivalently $-x_{l,i} - x_{l',j} - y_{t,i} - y_{t,j} \geq -3$. A pair $(\vec{V}, \vec{W})$ of clique and coclique falsifies L (in one of their matrix representations), iff i and j are in the clique and also i and j are in the same partition in the coclique. Then we define

$$C^L_{-2,-2} = e_{i,j}.$$

As required, $C^L_{-2,-2}$ outputs 1 on all the cliques with nodes i and j in the clique, and 0 on all the cocliques for which i and j are in the same partition. We only need one non trivial monotone circuit for L since $M = -2$, $N = -2$ is the only pair of values such that $M+N < -3$, that can be achieved.

Case 1: L was derived by a sound rule of inference from L_1, and L_2.
Say L_1 is $\sum d_{i,j}x_{i,j} + \sum e_{i,j}y_{i,j} \geq c_1$ and L_2 is $\sum f_{i,j}x_{i,j} + \sum g_{i,j}y_{i,j} \geq c_2$.

Let us first define two sets of pairs of integers: T_M, and T_N. T_M is defined by: $(M_1, M_2) \in T_M$ iff there exists a clique such that in some matrix representation $\vec{X}$,

$$\sum a_{i,j}X_{i,j} = M \ , \ \sum d_{i,j}X_{i,j} = M_1 \ , \ \sum f_{i,j}X_{i,j} = M_2$$

Similarly, T_N is defined by: $(N_1, N_2) \in T_N$ iff there is a coclique such that in some matrix representation $\vec{Y}$,

$$\sum b_{i,j}Y_{i,j} = N \ , \ \sum e_{i,j}Y_{i,j} = N_1 \ , \ \sum g_{i,j}Y_{i,j} = N_2$$

As visual aid, we will form a rectangular grid with the rows labeled with pairs $(M_1, M_2) \in T_M$, and the columns labeled with pairs $(N_1, N_2) \in T_N$. The existence of the entry $((M_1, M_2), (N_1, N_2))$ in the grid means that there exists a pair of clique and coclique $(\vec{V}, \vec{W})$, with matrix representation $(\vec{X}, \vec{Y})$, such that $\sum a_{i,j}X_{i,j} = M$, $\sum d_{i,j}X_{i,j} = M_1$, $\sum f_{i,j}X_{i,j} = M_2$, $\sum b_{i,j}Y_{i,j} = N$, $\sum e_{i,j}Y_{i,j} = N_1$, and $\sum g_{i,j}Y_{i,j} = N_2$. Since $M + N < c$, $(\vec{X}, \vec{Y})$ falsifies L. By soundness of the rule of inference used to derive L, $(\vec{X}, \vec{Y})$ has to either falsify L_1 or L_2. Now, if $(\vec{X}, \vec{Y})$ falsifies L_1 with $M_1 + N_1 < c_1$, by the induction hypothesis we have the monotone circuit $C^{L_1}_{M_1,N_1}$, that on input $\vec{V}$ gives output

1, and on input $\vec{W}$ gives output 0. Then in the entry $((M_1, M_2), (N_1, N_2))$ of the grid we write $C^{L_1}_{M_1,N_1}$. On the other hand, if $(\vec{X}, \vec{Y})$ doesn't falsify L_1, then it falsifies L_2, with $M_2 + N_2 < c_2$, by the induction hypothesis we have the monotone circuit $C^{L_2}_{M_2,N_2}$, that on input $\vec{V}$ outputs 1, and on input $\vec{W}$ outputs 0. Then the position $((M_1, M_2), (N_1, N_2))$ in the grid gets the circuit $C^{L_2}_{M_2,N_2}$.

With the visual aid of the grid, we can now describe the monotone circuit $C^{L}_{M,N}$: In each row take the AND of all the circuits ($C^{L_1}_{M_1,N_1}$, or $C^{L_2}_{M_2,N_2}$) in that row. After doing that take the OR of all of those ANDs.

Let us show that $C^{L}_{M,N}$ works. Suppose we have a pair of truth assignments $(\vec{V}, \vec{W})$ where $\vec{V}$ represents a clique and $\vec{W}$ represents a coclique, and for some matrix representation $(\vec{X}, \vec{Y})$ of $(\vec{V}, \vec{W})$, $\sum a_{i,j}X_{i,j} = M$ and $\sum b_{i,j}Y_{i,j} = N$. We need to show that $C^{L}_{M,N}$ on input $\vec{V}$ is 1 and on $\vec{W}$ is 0 :

To get 1 on input $\vec{V}$ we must get 1 in one of the AND-s, so that the OR is 1. Let M_1 be $\sum d_{i,j}X_{i,j}$, and M_2 be $\sum f_{i,j}X_{i,j}$. Then there is a row labeled with (M_1, M_2) on the grid. Because each circuit in that row gives 1 on $\vec{V}$, their ANDs also gives 1 on $\vec{V}$.

To get 0 on circuit $C^{L}_{M,N}$ with input $\vec{W}$, all the ANDs have to be 0 on $\vec{W}$. Thus in every AND, there must be a circuit that on $\vec{W}$ is 0. Let N_1 be $\sum e_{i,j}Y_{i,j}$, and N_2 be $\sum g_{i,j}Y_{i,j}$. Then there is a column labeled with (N_1, N_2) on the grid. Because each circuit on that column gives 0 on $\vec{W}$, there is a circuit on each row that gives 0 on $\vec{W}$. Thus all the ANDs will be 0 on $\vec{W}$.

Let us now check the extra work needed to build the circuits $C^{L}_{M,N}$ for all pairs (M, N). The grid we just described is of size $O(s^4)$. For a fixed pair (M, N), we compute $O(s^4)$ ANDs and afterwards, $O(s^2)$ ORs (we assume the fanin of all the gates in the circuit is 2). Also, there are $O(s^2)$ possible pairs. So for all pairs (M, N), the total extra work for L is $O(s^6)$. □

Corollary 3 *Let P be a Cutting Planes refutation of $\neg CLIQUE_{n,k}$, with $k = n^{2/3}$. For every $\epsilon < 1/3$, if all the coefficients in all the inequalities in P are smaller than $O(2^{n^\epsilon})$, then the length of P is $\Omega(2^{n^\epsilon})$.*

Corollary 4 *For every $\epsilon < 1/3$, the unary-representation-size of any Cutting Planes refutation of $\neg CLIQUE_{n,k}$, with $k = n^{2/3}$, is $\Omega(2^{n^\epsilon})$*

4.2 A Separation between Frege Systems and CP^*

The lower bounds given above show that Frege systems are strictly stronger than CP^*. This is because the clique tautology has a polynomial size Frege proof. Let us show this. We will do it by reducing $\neg CLIQUE_{n,k}$ to the negation of the pigeonhole principle. Since Buss [B] showed that the pigeonhole principle has polynomial size Frege proofs, $CLIQUE_{n,k}$ must also have polynomial size Frege proofs.

Let us show now the reduction of $\neg CLIQUE_{n,k}$ to the negation of the pigeonhole principle. For all i, $1 \le i \le k$, and all j, $1 \le j \le (k-1)$, we define

$$P_{i,j} = \bigvee_{l=1}^{n} (x_{i,l} \wedge y_{j,l})$$

For all i, $1 \le i \le k$, $P_{i,1} \vee P_{i,2} \vee \cdots \vee P_{i,(k-1)}$ is obtained from clauses 1-5. And $\neg P_{i,j} \vee \neg P_{l,j}$, for all i, l, j are obtained from clauses of type 6.

4.3 Bounds for Generalized Models

The above proof shows how to extract a monotone circuit from a proof, as long as the rules are sound, and the formulas are small-weight threshold formulas. With some modifications, this proof can be generalized to the setting where formulas have small communication complexity. The idea here is that the grid will now be partitioned into rectangles according to the communication protocol. Because the protocol is short, the number of rectangles is small, and therefore the final monotone circuit will have small size. This leads to our main theorem.

Theorem 5 *Given any Inference refutation (see section 2.2) of $\neg CLIQUE_{n,k}$, we can build a monotone circuit of size $\le m \cdot 2^{3D+1}$, for some clique separator function $Q_{n,k}$. Here m is the length of the refutation, and D is an upper bound for the communication complexity of all the formulas in the refutation.*

Proof Again, we build the circuit by levels. At the level corresponding to line L, the circuit only distinguishes pairs of cliques and cocliques that in some matrix representation falsify the line L. The final formula in the refutation is unsatisfiable. Since every pair of clique and coclique in matrix representation falsifies it, the circuit at that level will compute the clique function on all k-cliques and all $(k-1)$-cocliques.

Give the variables $x = \{x_{i,j}\}$ to Player I, and the variables $y = \{y_{i,j}\}$ to Player II. The communication complexity of all the functions below is considered in respect to this particular partition. The communication complexity of any formula L in the proof (with respect to this partition) is at most D. Let P_L be one fixed communication protocol for determining the truth value of L (with communication complexity $\le D$). For the last line of the refutation (which is unsatisfiable), take P_L to be the trivial protocol.

Let H be the set of all boolean strings of length $\le D$. We will call H **the set of all possible histories**. For the truth assignment $\vec{X}, \vec{Y}$ (for the variables x, y), define $h_L(\vec{X}, \vec{Y}) \in H$ to be the string communicated by P_L on $\vec{X}, \vec{Y}$. We call $h_L(\vec{X}, \vec{Y})$; **the history of** $(\vec{X}, \vec{Y})$. Since P_L can be viewed also as a communication protocol for the function h_L, the communication complexity of the function h_L is at most D.

Clearly, the value of the history $h = h_L(\vec{X}, \vec{Y})$ determines the value of the formula L on $(\vec{X}, \vec{Y})$. We denote this value by $L(h)$. $L(h)$ is undefined if the history h cannot be the communication string of the protocol P_L. If $L(h)$ is $FALSE$, we say that the history h falsifies the formula L. If $L(h)$ is $TRUE$, we say that the history h satisfies the formula L. For the last line of the refutation, since P_L is the trivial protocol, $h_L(\vec{X}, \vec{Y})$ is always the empty string, and for the empty string h, $L(h)$ is $FALSE$ (since the last line is unsatisfiable).

For any line L in the refutation, and any history $h \in H$, which falsifies L, we will build a monotone circuit C_h^L. To build the circuits C_h^L, we will use circuits $C_{h'}^{L'}$ from previous levels. Let $(\vec{V}, \vec{W})$ be a pair of truth assignments for the input variables $e_{i,j}$ (of the circuits that we are building), such that $\vec{V}$ represents a k-clique, and $\vec{W}$ represents a $(k-1)$-coclique. The circuits C_h^L will satisfy the following :

1. If some matrix representation $(\vec{X}, \vec{Y})$ of $(\vec{V}, \vec{W})$ satisfies $h_L(\vec{X}, \vec{Y}) = h$ then C_h^L on input $\vec{V}$ gives output 1 and C_h^L on input $\vec{W}$ gives output 0.
2. The extra work that needed to build all the circuits C_h^L (for all h), from all the circuits at previous levels, is at most 2^{3D+1}.

Clearly, for the last line of the refutation (which is unsatisfiable), the circuit C_h^L (for the empty string h) will compute the clique function on all k-cliques and all $(k-1)$-cocliques, and the circuit size will be at most $m \cdot 2^{3D+1}$.

Assume that L is the l-th formula in the refutation. Again, we will build the circuits C_h^L by induction on l. Suppose that for every line, L', numbered $< l$ we have the circuits $C_{h'}^{L'}$. We will now build the circuits for the l-th line. As before we divide the proof into two cases:

Case 0: L is an axiom of the types $1-6$.

If L is of the types $1-5$, all pairs of clique and coclique $(\vec{V}, \vec{W})$, in any matrix representation, satisfy the line L, and therefore no circuit C_h^L is needed.

If L is of type 6 then a pair $(\vec{V}, \vec{W})$ of clique and coclique falsifies L (in one of their matrix representations), iff i and j are in the clique and also i and j are in the same part of the coclique. Then we define for all $h \in H$ that falsifies L :

$$C_h^L = e_{i,j}$$

As required, C_h^L gives output 1 on all the cliques with nodes i and j in the clique, and 0 on all the cocliques for which i and j are in the same part.

Case 1: L was derived by a sound rule of inference from L_1, and L_2.

The proof follows easily from the following lemma:

Lemma 6 *Let R_X be a set of truth assignments for x, and R_Y a set of truth assignments for y. Assume that every $(\vec{X}, \vec{Y}) \in R_X \times R_Y$ falsifies L, and that in the rectangle $R = R_X \times R_Y$ the function $\bar{h}$ defined as*

$$\bar{h}(\vec{X}, \vec{Y}) = \left(h_{L_1}(\vec{X}, \vec{Y}), h_{L_2}(\vec{X}, \vec{Y})\right)$$

has communication complexity d. Then there exists a monotone circuit C_R such that:

1. *C_R uses the circuits $\{C_{h'}^{L_1}\}, \{C_{h'}^{L_2}\}$ (for all h'), plus $2^d - 1$ extra gates. i.e. the extra work needed to build C_R is $2^d - 1$.*
2. *If some matrix representation $(\vec{X}, \vec{Y})$ of $(\vec{V}, \vec{W})$ satisfies $(\vec{X}, \vec{Y}) \in R$ then C_R on input $\vec{V}$ gives output 1 and C_R on input $\vec{W}$ gives output 0.*

Proof The proof is by induction on d: For $d = 0$, $\bar{h}$ is known at the beginning, i.e $h_1 = h_{L_1}(\vec{X}, \vec{Y})$, $h_2 = h_{L_2}(\vec{X}, \vec{Y})$ are fixed on the rectangle R. Since the rectangle R falsifies L, and since L was derived by a sound rule from L_1, L_2, we know that either h_1 falsifies L_1, or h_2 falsifies L_2. w.l.o.g. h_1 falsifies L_1. Then the required circuit is

$$C_R = C_{h_1}^{L_1}$$

Clearly (by the induction hypothesis for $C_{h_1}^{L_1}$) if some matrix representation $(\vec{X}, \vec{Y})$ of $(\vec{V}, \vec{W})$ satisfies $(\vec{X}, \vec{Y}) \in R$ then $C_{h_1}^{L_1}$ on input $\vec{V}$ gives output 1 and $C_{h_1}^{L_1}$ on input $\vec{W}$ gives output 0.

For $d > 0$ we look at the communication protocol for $\bar{h}$, in the rectangle R. We have two cases:

Case 1: Player I sends the first bit: Let R_0 be the subset of R_X, where this bit is 0, and let R_1 be the subset of R_X where this bit is 1. The rectangles $R_0 \times R_Y$, and $R_1 \times R_Y$ satisfy the induction hypothesis for $d-1$. The circuit C_R in this case will be

$$C_R = C_{R_0 \times R_Y} \vee C_{R_1 \times R_Y}$$

Clearly, by the induction hypothesis C_R works, and its size is $2 \cdot (2^{d-1} - 1) + 1 = 2^d - 1$, as required.

Case 2: Player II sends the first bit: Let R_0 be the subset of R_Y, where this bit is 0, and let R_1 be the subset of R_Y where this bit is 1. The rectangles $R_X \times R_0$, and $R_X \times R_1$ satisfy the induction hypothesis for $d-1$. The circuit C_R in this case will be

$$C_R = C_{R_X \times R_0} \wedge C_{R_X \times R_1}$$

Clearly, by the induction hypothesis C_R works, and its size is $2 \cdot (2^{d-1} - 1) + 1 = 2^d - 1$, as required. □

A standard argument shows that for a string h, the set of pairs $\vec{X}, \vec{Y}$, with the history $h_L(\vec{X}, \vec{Y}) = h$, is in fact a rectangle $R = R_X \times R_Y$. Assume that h falsifies L. Since the communication complexity of the function $\bar{h}(\vec{X}, \vec{Y}) = (h_{L_1}(\vec{X}, \vec{Y}), h_{L_2}(\vec{X}, \vec{Y}))$ is always smaller than $2D$, the conditions of the lemma are satisfied with $d = 2D$. Thus we can use the lemma to build C_h^L.

Since there are at most 2^{D+1} possible histories, the extra work needed to build all the circuits C_h^L is at most 2^{3D+1}. □

Corollary 7 *Let P be any Inference refutation of $\neg CLIQUE_{n,k}$, with $k = n^{2/3}$. For every $\epsilon < 1/3$, if the communication complexity of every formula in P is smaller than $O(n^\epsilon)$, then the length of P is $\Omega(2^{n^\epsilon})$.*

Corollary 8 *For every $\epsilon < 1/3$, If P is a Generalized CP^* refutation of $\neg CLIQUE_{n,k}$ then P must have size of $\Omega(2^{n^\epsilon})$.*

Our lower bound for CP^* also holds if we add the deduction rule:

Theorem 9 *Let P be a refutation of $\neg CLIQUE_{n,k}$ in CP^* with deduction. Then for every $\epsilon < 1/3$, the length of P is $\Omega(2^{n^\epsilon})$.*

Proof The proof is by a reduction to the previous corollaries. Consider the following abstract system: The formulas of the system are of the type $p \to L$, where L is a CP^* formula, and p is a set of CP^* formulas. The formula $p \to L$ can be concluded from $p_1 \to L_1$, $p_2 \to L_2$ iff any truth assignment that falsifies $p \to L$ also falsifies either $p_1 \to L_1$ or $p_2 \to L_2$ (i.e. any sound inference). Clearly this general rule is stronger than the following two simpler rules:

1. If L can be concluded from L_1, L_2 by any sound rule then $p \to L$ can be concluded from $p \to L_1$, $p \to L_2$.
2. for any CP^* formula L, the formula $p \to 0 \geq 1$ can be concluded from $p \vee \{L\} \to 0 \geq 1$, $p \vee \{\neg L\} \to 0 \geq 1$.

Therefore this model can simulate CP^* with deduction: We simulate the proofs in part two (of a CP^* with deduction proof), by using inferences of type 1, and we finally prove $0 \geq 1$ using inferences of type 2. In the simulation the set p will be the set of all the formulas that we assume at some point. Given a proof in CP^* with deduction, we translate it into a proof in the new system. If the tree of deductions in the original proof is of size bigger than $O(2^{n^\epsilon})$ then we are done. Otherwise, since the tree is balanced, the size of the set p in each formula in the translation will be smaller than $O(n^\epsilon)$.

Now the only thing to see is that this falls into the category of generalized CP^*. To see this, one have to see that there is a short communication complexity protocol to decide whether $p \to L$ (soundness is clear by definition). This is true, because there is a short protocol for each of the formulas in p. So the two players can find out the value of each formula in p, and the value of the formula L, and then decide on the value of $p \to L$. The communiction complexity of this protocol is $O(n^{\epsilon'})$, for any $\epsilon < \epsilon' < 1/3$. □

5 Separation theorems

Informally, our version of the st-connectivity principle states that if G_x is a graph on n vertices, such that G_x consists of a single path of length l connecting vertex s to vertex t, and if G_y is a graph such that the vertices are partitioned into two sets (with s in one set and t in the other), and all edges are present within each set, but no edges are present between the sets, then there must be an edge present in G_x that is absent in G_y. The graph G_x is said to be a minterm of the st-connectivity function because G_x contains a path from s to t, but if any edge is taken away, this condition is violated. Similarly, the graph G_y is a maxterm of the st-connectivity function because G_y does not contain a path from s to t, but if any edge is added, then there will be a path from s to t.

We will formalize the negation of the st-connectivity principle for length l on graphs with n vertices by the propositional formula $\neg STCONN_n^l$. The underlying variables are $x = \{x_{i,j} \mid 1 \leq i \leq l, 1 \leq j \leq n\}$, and $y = \{y_{i,j} \mid i = 1,2,\ 1 \leq j \leq n\}$. The matrix x describes the graph G_x in the following way: the variable $x_{i,j}$ is 1 if and only if j is the i^{th} element on the path from s to t. Similarly, y describes the graph G_y, where $y_{1,j}$ is 1 iff vertex j is in set 1, and $y_{2,j}$ is 1 iff vertex j is in set 2.

The unsatisfiable formula $\neg STCONN_n^l$ is the conjunction of the following clauses. The clauses in (1)-(4) describe the condition that x must be a matrix that describes a path of length l from vertex 1 ($= s$) to vertex n ($= t$). The clauses in (5)-(7) say that y must be a partition of the vertices $[1, n]$ into two groups, where vertex 1 is in group 1 and vertex n is in group 2. The clauses in (8)-(9) say that if there is an edge from vertex i to vertex j in G_x, then i and j cannot be in different groups in G_y.

1. $x_{1,1}$; $x_{l,n}$
2. $\bigvee_{k=1}^{n} x_{i,k}$ for all i, $1 \leq i \leq l$
3. $\neg x_{i,j} \vee \neg x_{i,k}$ for all i, j, k such that $1 \leq i \leq l$, $1 \leq j, k \leq n$ and $j \neq k$
4. $\neg x_{i,k} \vee \neg x_{j,k}$ for all i, j, k such that $1 \leq i, j \leq l$, $i \neq j$ and $1 \leq k \leq n$
5. $y_{1,1}$; $y_{2,n}$; $\neg y_{2,1}$; $\neg y_{1,n}$
6. $y_{1,i} \vee y_{2,i}$ for all i, $1 \leq i \leq n$
7. $\neg y_{1,i} \vee \neg y_{2,i}$ for all i, $1 \leq i \leq n$
8. $\neg x_{q,i} \vee \neg x_{q+1,j} \vee \neg y_{1,i} \vee \neg y_{2,j}$ for all q, i, j such that $1 \leq q \leq l-1$ and $1 \leq i, j \leq n$
9. $\neg x_{q,i} \vee \neg x_{q+1,j} \vee \neg y_{2,i} \vee \neg y_{1,j}$ for all q, i, j such that $1 \leq q \leq l-1$ and $1 \leq i, j \leq n$

The st-connectivity tautology described above will be used to separate CP^* from tree-like CP^*, and also to separate bounded-depth Frege from tree-like CP^*. First we will show that $STCONN_n^l$ has short and natural bounded-depth Frege proofs. Next we will show that $STCONN_n^l$ also has short proofs in CP^*. This is not as obvious as the bounded-depth Frege proof, but follows along similar lines. Lastly, we derive lower bounds for tree-like CP^* proofs of $STCONN_n^l$, using the method in [IPU].

5.1 Bounded-depth Frege proofs of $STCONN_n^l$

Small size bounded-depth Frege refutations of $\neg STCONN_n^l$ are quite natural. First, for all q, $1 \leq q \leq l-1$, and all i, j, $1 \leq i, j \leq n$, we obtain from clauses 8 and 9 the formulas:

$$x_{q,i} \wedge x_{q+1,j} \to ((y_{1,i} \wedge y_{1,j}) \vee (y_{2,i} \wedge y_{2,j})).$$

These formulas express the fact that if there is a path of length 1 from i to j in G_x, then i and j must be in the same group in G_y. Now using these formulas, we can derive the following formulas for all k, $1 \leq k \leq n$, $k \neq i$, $k \neq j$,

$$(x_{q,i} \wedge x_{q+1,k} \wedge x_{q+2,j}) \to ((y_{1,i} \wedge y_{1,j}) \vee (y_{2,i} \wedge y_{2,j})).$$

The above formulas express the fact that if there is a path of length 2 from i to j through vertex k in G_x, then i and j must be in the same group in G_y. Combining these formulas, we then obtain:

$$(x_{q,i} \wedge x_{q+2,j} \wedge (\bigvee_{k=1}^{n} x_{q+1,k})) \to ((y_{1,i} \wedge y_{1,j}) \vee (y_{2,i} \wedge y_{2,j})).$$

From the above formula together with the initial clause, $\bigvee_{k=1}^{n} x_{q+1,k}$ (from 2), we can now derive

$$(x_{q,i} \wedge x_{q+2,j}) \rightarrow ((y_{1,i} \wedge y_{1,j}) \vee (y_{2,i} \wedge y_{2,j})).$$

This formula expresses the fact that if there is a path of length 2 from i to j in G_x, then i and j must be in the same group in G_y. Repeating this argument $l-1$ times, we can eventually derive

$$(x_{1,1} \wedge x_{l,n}) \rightarrow (y_{1,1} \wedge y_{1,n}) \vee (y_{2,1} \wedge y_{2,n}).$$

Using the initial clauses in 1, $x_{1,1}$, $x_{l,n}$, we then derive $(y_{1,1} \wedge y_{1,n}) \vee (y_{2,1} \wedge y_{2,n})$. But now it is easy to derive false using the initial clauses from 5.

5.2 Small-weight non-tree-like CP proofs of $STCONN_n^l$

First we must convert the above clauses expressing $\neg STCONN_n^l$ into inequalities. Although these translations are simple, we describe them below.

1. $x_{1,1} \geq 1$; $x_{l,n} \geq 1$
2. $\sum_{k=1}^{n} x_{i,k} \geq 1$ for all i, $1 \leq i \leq l$
3. $1 \geq x_{i,j} + x_{i,k}$ for all i,j,k such that $1 \leq i \leq l$, $1 \leq j,k \leq n$, $j \neq k$
4. $1 \geq x_{i,k} + x_{j,k}$ for all i,j,k such that $1 \leq i,j \leq l$, $1 \leq k \leq n$ and $i \neq j$
5. $y_{1,1} \geq 1$; $y_{2,n} \geq 1$; $0 \geq y_{2,1}$; $0 \geq y_{1,n}$
6. $y_{1,i} + y_{2,i} \geq 1$ for all i such that $1 \leq i \leq n$
7. $1 \geq y_{1,i} + y_{2,i}$ for all i, $1 \leq i \leq n$
8. $3 \geq x_{q,i} + x_{q+1,j} + y_{1,i} + y_{2,j}$ for all q,i,j, $1 \leq q \leq l-1$, $1 \leq i,j \leq n$
9. $3 \geq x_{q,i} + x_{q+1,j} + y_{2,i} + y_{1,j}$ for all q,i,j, $1 \leq q \leq l-1$, $1 \leq i,j \leq n$

In addition to the above equations, we also have the inequalities $0 \leq x_{i,j}$, $0 \leq y_{i,j}$, $y_{i,j} \leq 1$ and $x_{i,j} \leq 1$ for all variables in the formula.

We will now describe a small-weight Cutting Planes refutation of the above inequalities. For each a, $0 \leq a \leq l$, we will derive the inequalities:

$$3 \geq x_{q,i} + x_{q+a,k} + y_{1,i} + y_{2,k}$$

for all q,i,k such that $q+a \leq l$, $1 \leq i,k \leq n$. Furthermore, the size of these derivations will be polynomial, and the weights will all be bounded by a polynomial. These formulas intuitively express the fact that if i and k are connected by a path of length a in G_x, then they must be in the same set in G_y. When $a = l-1$, $q = 1$, $i = 1$ and $k = n$, we have

$$3 \geq x_{1,1} + x_{l,n} + y_{1,1} + y_{2,n},$$

which contradicts clauses (1) and (5).

It is thus left to derive the inequalities: $3 \geq x_{q,i} + x_{q+a,k} + y_{1,i} + y_{2,k}$, for all a, $1 \leq q+a \leq l$, $1 \leq i,k \leq n$. The base case, when $a = 1$, are initial inequalities, so there is nothing to prove. By the induction hypothesis, we have small-weight, polynomial size derivations for: $3 \geq x_{q,i} + x_{q+a,k} + y_{1,i} + y_{2,k}$, and for $3 \geq x_{q+a,k} + x_{q+a+1,j} + y_{1,k} + y_{2,j}$, for all $1 \leq k \leq n$. Adding these two formulas we obtain for each value of k:

$$6 \geq x_{q,i} + 2x_{q+a,k} + x_{q+a+1,j} + y_{1,i} + y_{2,k} + y_{1,k} + y_{2,j}.$$

Now for each k, adding to the above the initial inequality, $y_{1,k} + y_{2,k} \geq 1$, we obtain for each k (a): $5 \geq 2x_{q+a,k} + Z$, where

$$Z = x_{q,i} + x_{q+a+1,j} + y_{1,i} + y_{2,j}.$$

Now separately, we can derive (b): $4 \geq Z$ from the initial inequalities $1 \geq x_{q,i}$, $1 \geq x_{q+a+1,j}$, $1 \geq y_{1,i}$ and $1 \geq y_{2,j}$. Adding (a) and (b) and dividing by 2, we then obtain for each k, $4 \geq x_{q+a,k} + Z$. Adding these n equations together, one for each k, we obtain

$$4n \geq nZ + \sum_{k=1}^{n} x_{q+a,k}.$$

However, $1 \leq \sum_{k=1}^{n} x_{q+a,k}$ is an initial inequality, and thus we can add this and divide by n to obtain the desired inequality $3 \geq Z$.

The above derivation has polynomial size, and the weights have size $O(n)$. Note that our Cutting Planes proof is not tree-like because the intermediate formulas talking about paths of length a must be used many times in order to generate the intermediate formulas talking about paths of length $a+1$.

5.3 Lower bounds for small-weight, tree-like CP proofs of $STCONN_n^l$

In this section we will prove the following theorem.

Theorem 10 *For $l = n^{1/100}$, any tree-like CP^* refutation of $\neg STCONN_n^l$ requires superpolynomial size.*

The above theorem is an application of the method in [IPU].

Corollary 11 *Bounded-depth Frege cannot be p-simulated by small-weight, tree-like Cutting Planes.*

Corollary 12 *Small-weight Cutting Planes cannot be p-simulated by small-weight tree-like Cutting Planes.*

We remark that it is already known that Cutting Planes cannot be p-simulated by Bounded-depth Frege, because the propositional pigeonhole principle has short CP proofs, but requires exponential-size bounded-depth Frege proofs. Actually, the short proof of the pigeonhole principle is in tree-like CP^*. This means that Bounded-depth Frege and tree-like CP^* are incomparable.

We will now describe the proof of theorem 10. The communication complexity problem associated with $STCONN_n^l$ is as follows. Let M_x be the set of graphs on n vertices that contain a path of length l from s to t and no other edges, and let M_y be the set of graphs on n vertices that consist of a partition of the vertices into two sets B_s and B_t such that $s \in B_s$, $t \in B_t$, and all edges within B_s and within B_t are present and no other edges are present. The communication complexity problem, $Findedge(l,n)$ is: Player I is given a graph $G_x \in M_x$, and Player II is given a graph $G_y \in M_y$, and they want to find an edge of G_x which is not present in G_y. Let $C_\epsilon(Findedge(l,n))$ denote the ϵ-probabilistic communication complexity of the game $Findedge(l,n)$.

Lemma 13 *Let P be a tree-like CP^* refutation of $\neg STCONN_n^l$ with at most n^k lines, for some constant k. Then for $\epsilon \leq 1/4$, $C_\epsilon(Findedge(l,n)) \leq O(\log n(\log\log n)^2)$.*

Proof The proof is implicit in [IPU]. The method described there implies that any CP proof for $\neg STCONN_n^l$ gives a probabilistic communication protocol for the game $Findedge(l,n)$, with complexity $O(\log m(\log\log(sn))^2)$, where m is the length of the proof, and s is an upper bound for the largest coeeficient involved. □

To complete the proof of the lower bound, we will need the following lower bound due to Raz and Wigderson [RW1] (see also [Raz], and [BL]).

Theorem 14 *Let $\epsilon \leq 1/4$ and let $l = n^{1/100}$. Then for sufficiently large n, $C_\epsilon(Findedge(l,n)) \geq \Omega((\log n)^2)$.*

Theorem 10 now follows from the above theorem and the above lemma.

6 Conclusions

The proofs in this paper are actually instances of a more general interpolation theorem for small-weight Cutting Planes (this possibility was brought to our attention by Jan Krajicek as well as Russell Impagliazzo, and was recently proved in [K]). This theorem states, roughly, that if $\{A_i(x,y),\ B_i(x,z) \mid i \leq q\}$ are a set of unsatisfiable clauses with a polynomial-size CP^* refutation, and such that: $A_i(x,y)$ involve variables $x_1,..,x_n$ and $y_1,..,y_n$ and $B_i(x,z)$ involve variables $x_1,..,x_n$ and $z_1,..,z_n$, then there exists an "interpolant" formula, $C(x)$ such that: $C(x)$ is computable by a polynomial-size circuit, and $C(x)$ outputs 1 whenever there exists a y such that all of the clauses $A_i(x,y)$ are true, and outputs 0 whenever there exists a z such that all of the clauses $B_i(x,z)$ are true. In special cases, the formula $C(x)$ can also be shown to be computable by a monotone polynomial-size circuit, as is the situation for the tautologies discussed in this paper. This general interpolation theorem can be proven using our method and moreover, it can be applied to extend a result of Razborov [Razb2]. Namely, we can show that a propositional statement expressing "Satisfiability is not computable by polynomial-size boolean circuits" requires large CP^* proofs, assuming a standard cryptographic assumption.

One obvious open problem is to extend our lower bound to the Cutting Planes proof system with unrestricted weights.

Acknowledgements

The authors would like to thank Russell Impagliazzo, Jan Krajicek and Sam Buss for very helpful conversations.

References

[AB] Alon N., and Boppana R., "The Monotone Circuit Complexity of Boolean Functions", *Combinatorica*, Vol 7, No. 1, pp. 1-22, 1987.

[Ajt] M. Ajtai, "The complexity of the pigeonhole principle," forthcoming. Preliminary version, *29th Annual Symposium on the Foundations of Computer Science*, pp. 346-355, 1988.

[B] Buss, S. "Polynomial size proofs of the propositional pigeonhole principle," *Journal of Symbolic Logic*, v. 52 (1987), pp. 916-927.

[BIKPPW] Beame, P., Impagliazzo, R., Krajicek, J., Pitassi, T., Pudlak, P., Woods, A., "Exponential lower bounds for the pigeonhole principle," *Symposium on Theoretical Computer Science*, 1992.

[BL] Beame, P., Lawry, J., "Randomized versus Nondeterministic Communication Complexity," *Symposium on Theoretical Computer Science*, 1992.

[C] Clote P., "Cutting planes and constant depth Frege proofs," Manuscript 1993.

[CCT] Cook, W., Coullard, C.R., Turan, G., "On the complexity of cutting plane proofs," *Discrete Applied Mathematics*, 18, 1987, pp. 25-38.

[CR] Cook, S., Reckhow, R., "The relative efficiency of propositional proof systems," *Journal of Symbolic Logic*, 44, 1979.

[Chv] Chvatal V., "Edmond Polytopes and a Hierarchy of Combinatorial Problems", *Discrete Math.*, 4, pp. 305-337, 1973

[G] Goerdt, A. "Cutting plane versus Frege proof systems," *Lecture Notes in Computer Science*, 533.

[Gom] Gomory R. E., "An Algorithm for Integer Solutions of Linear Programs" *in: R.L. Graves and P. Wolfe, eds., Recent Advances in Mathematical Programming*, McGraw-Hill, New York, pp. 269-302, 1963

[H] Haken, A. "The intractability of Resolution," *Theoretical Computer Science*, 39, 1985. pp. 297-308.

[IPU] Impagliazzo, R., Pitassi, T., Urquhart, A., "Upper and lower bounds for tree-like Cutting Planes proofs," Proceedings from Logic in Computer Science, 1994.

[K] Krajicek J., "Interpolation theorems, lower bounds for proof systems and independence results for bounded arithmetic", manuscript.

[KW] Karchmer M., Wigderson A., "Monotone Circuits for Connectivity Require Super-logarithmic Depth" *Proceedings of the 20th STOC*, pp. 539-550, 1988.

[RW1] Raz R., Wigderson A., "Probabilistic Communication Complexity of Boolean Relations", *Proc. of the 30th FOCS*, 1989.

[RW2] Raz, R., Wigderson, A., "Monotone circuits for matching require linear depth," *ACM Symposium on Theory of Computing*, 1990, pp. 287-292.

[Raz] Raz R., "Lower Bounds for Probabilistic Communication Complexity and for the depth of Monotone Boolean Circuits", Phd thesis, The Hebrew University, 1992, (in Hebrew)

[Razb1] Razborov A., "Lower Bounds for the Monotone Complexity of some Boolean Functions", *Dokl. Ak. Nauk. SSSR*, Vol 281, pp. 798-801, 1985 (in Russian). English translation in *Sov. Math. Dokl.* Vol 31, pp. 354-357, 1985.

[Razb2] Razborov, A., "Unprovability of lower bounds on the circuit size in certain fragments of bounded arithmetic," preprint, March 3, 1994.

[RR] Razborov A., Rudich S., "Natural Proofs" *Symposium on Theoretical Computer Science*, 1993.