Upper and Lower Bounds for Tree-like Cutting Planes Proofs

Russell Impagliazzo* — UCSD
Toniann Pitassi† — UCSD and U. of Pittsburgh
Alasdair Urquhart‡ — University of Toronto

Abstract

In this paper we study the complexity of Cutting Planes (CP) refutations, and tree-like CP refutations. Tree-like CP proofs are natural and still quite powerful. In particular, the propositional pigeonhole principle (PHP) has been shown to have polynomial-sized tree-like CP proofs. Our main result shows that a family of tautologies, introduced in this paper requires exponential-sized tree-like CP proofs. We obtain this result by introducing a new method which relates the size of a CP refutation to the communication complexity of a related search problem. Because these tautologies have polynomial-sized Frege proofs, it follows that tree-like CP cannot polynomially simulate Frege systems.

1 Introduction

An important open problem is to determine whether there exists a propositional proof system that admits short (polynomial size) proofs for all tautologies, or equivalently, whether or not NP equals coNP. In order to attack

*Research supported by NSF NYI grant CCR-92-570979
†Research supported by NSF fellowship and UC Presidential fellowhip
‡Research supported by the Natural Sciences and Engineering Research Council of Canada

this difficult problem, many researchers have studied the complexity of particular proof systems. To date, the strongest superpolynomial lower bounds known are for Frege systems where the depth of each formula is constant (i.e., bounded-depth Frege systems.) Clearly the next big step would be to prove superpolynomial lower bounds for unrestricted Frege systems. This problem appears to be very hard, and moreover, there are very few natural examples of tautologies which are believed to require superpolynomial-sized Frege proofs. In this paper we study the complexity of *Cutting Planes* proofs — a proof system which cannot be simulated efficiently by bounded-depth Frege systems, but still appears to be weaker than unrestricted Frege systems.

The Cutting Planes system (CP) is a refutation system for propositional logic, based on showing that there are no integer solutions for a family of linear inequalities associated with an unsatisfiable propositional formula. This system was introduced in [CCT], and shown to be a sound and complete proof system for all unsatisfiable formulas in conjunctive normal form. For the purposes of this paper, we can think of Cutting Planes as a particular complete proof system where each formula is a linear inequality, and all of the rules have the property that they derive one inequality from two antecedents, and the rule is sound. A *tree-like* Cutting Planes proof is a special case of a Cutting Planes proof, where the underlying directed acyclic graph is a tree. Clearly, unrestricted Cutting Planes refutations are at least as powerful as tree-like refutations, and it is currently not known if they are more powerful.

Tree-like Cutting Planes proofs are natural and non-trivial. In particular, the propositional pigeonhole principle (PHP) has been shown to have polynomial-sized tree-like CP proofs [CCT]. This tautology is important since almost all non-trivial lower bounds for size of proofs have used PHP as the family of hard examples. The main result of this paper is an exponential lower bound on the size of tree-like Cutting Planes proofs for certain tautologies. Since our tautologies have polynomial-sized tree-like Frege proofs, this shows that tree-like Cutting Planes systems cannot p-simulate tree-like Frege systems. (More precisely, we show that tree-like Cutting Planes cannot p-simulate tree-like bounded-depth Frege+PHP.)

In section 2, we review the Cutting Planes proof system, and what is known about the complexity of CP proofs. We conclude section 2 with a discussion of what can and cannot be proven by polynomial-sized Cutting Planes proof systems. We give a new, short CP proof for a general class of tautologies which includes the PHP as a special case, and then discuss hard examples for CP. In section 3, we prove our exponential lower bound and in section 4, we discuss the corresponding upper bound. Finally, we conclude in section 5 with open problems and conjectures concerning lower bounds for Cutting Planes and Frege systems.

2 Definitions and background

2.1 Cutting Planes proofs

We first review the Cutting Planes (CP) refutation system. For a more complete treatment, see [C1], [G1], or [CCT]. Cutting Planes formulae are inequalities of the form

$$a_1x_1 \ + \ a_2x_2 \ + \ ...a_nx_n \ \geq \ A,$$

where $a_i, A \in Z$. The Cutting Planes system has four sound rules of inference: (1) basic algebraic simplification of sums and products of integers and integers; (2) addition of two inequalities; (3) multiplication of an inequality by an integer; and (4) division: if c divides each a_i, then we can derive $\sum_i a_i/cx_i \geq \lceil A/c \rceil$ from $\sum_i a_ix_i \geq A$.

In order to use the Cutting Planes system as a refutation system for CNF formulas, we must first translate CNF formulas into a system of linear inequalities. Define $R(x) = x$, and $R(\neg x) = 1 - x$. A clause $(l_1 \vee l_2 \vee .. \vee l_k)$ is translated into the linear inequality $\sum_{i=1}^k R(l_i) \geq 1$. The system of linear inequalities $E(f)$ corresponding to a CNF formula, f, is the set of inequalities we obtain by translating each clause in f, together with the inequalities $x \geq 0$ and $-x \geq -1$ for all variables x in f.

A Cutting Planes refutation of a CNF formula f is defined to be a directed acyclic graph where each node is labelled with a particular linear inequality, and such that: (1) each leaf formula is a linear inequality in $E(f)$; (2) intermediate formulas follow from two previous inequalities by one of the Cutting Planes rules, and the final (root) inequality is $0 \geq 1$. The *size* of a CP proof is equal to the number of formulas in the proof. A *tree-like* CP proof is a Cutting Planes proof where the underlying directed acyclic graph is a tree. Clearly, tree-like proofs are a special case of Cutting Planes proofs, and it is not known if they are strictly weaker.

We would like to point out that it is sensible to count the number of formulas rather than the *symbol size*. (The symbol size of a proof is the sum of the sizes of all formulas in the proof.) It was shown in [CCT] that any CP proof can be converted into a new CP proof such that the length of all of the coefficients in the new proof are polynomially bounded, and the number of formulas in the new proof is polynomial in the original number of formulas. Therefore, a tautology has a CP proof of polynomial symbol size if and only if the tautology has a CP proof with polynomially many formulas.

2.2 The complexity of Cutting Planes

In this section we review the *relative* complexity of the Cutting Planes proof system. We say that proof system A *p-simulates* proof system B if there exists a polynomial p such that: for every proof of size s in B, there is a corresponding proof of the same tautology in A with size at most $p(s)$. Two proof systems are *polynomially equivalent* if they p-simulate one another.

How powerful are Cutting Planes and tree-like Cutting Planes proof systems? It is straightforward to verify that CP is a generalization of Resolution—i.e. any Resolution proof can be polynomially-simulated by CP. Moreover, there are polynomial-sized CP proofs of the propositional PHP, but Haken [H] has shown that Resolution proofs of PHP require exponential size. Thus, Resolution cannot p-simulate CP.

It was shown in [G1] that any Cutting Planes proof can be p-simulated by a Frege proof. This result is not surprising because each formula in a CP refutation is just a depth-1 threshold formula, whereas *any* polynomial-sized formula may appear in a polynomial-sized Frege proof. On the other hand, because many polynomial-sized functions cannot be computed by depth-1 threshold formulas, (the parity function, for example) we expect that Cutting Planes cannot p-simulate Frege systems. Moreover, it can also be shown that tree-like CP lies between tree-like Resolution and Frege. (We do not distinguish between tree-like Frege and Frege because they have been shown to be polynomially equivalent [K].)

How does Cutting Planes compare with bounded-depth Frege systems? Recent exponential lower bounds for bounded-depth Frege proofs of PHP [BIK] combined with the upper bound for tree-like CP proofs of PHP [CCT] establish that bounded-depth Frege cannot p-simulate Cutting Planes or even tree-like Cutting Planes. Does CP p-simulate bounded-depth Frege systems? We expect the answer to be negative, but this is still an open problem.

Moreover, it may seem surprising that although there are strong lower bounds known for depth-1 threshold formulas, there are no lower bounds known for Cutting Planes, or even for tree-like Cutting Planes proofs prior to this paper. (In fact, the only lower bound for CP that we know of, prior to this paper, is a lower bound on a restricted CP system, due to Goerdt [G2].)

2.2.1 Easy and Hard Examples

As mentioned earlier, the canonical family of hard examples, PHP, has short CP proofs [CCT]. So we must first come up with new hard examples. In order to give the reader some intuition for Cutting Planes proofs, we will discuss several possibilities. As a first attempt, we consider various tautologies based on the mod 2 principle. (The mod 2 principle states that the size of a set cannot be congruent to both 0 and 1 mod 2.)

Let G be a graph with n vertices which does not contain a perfect matching. Then the "no perfect matching" principle for a graph $G = (V, E)$, $|V| = n$, states that any subgraph, $G' = (V, E')$, $E' \subset E$ of G has the property that some vertex has degree not equal to 1. The negation of this principle, $\neg NPM_G$ for $G = (V, E)$ can be expressed as the conjunction of the following clauses in the variables P_e, $e \in E$: $(\bigvee_{e, v \in e}\{P_e\})$, for all $v \in V$, and $(\neg P_e \vee \neg P_f)$, for all $e, f \in E$, $e \cap f \neq \emptyset$. For *any* graph G, there is a corresponding $\neg NPM_G$ formula, but this formula is unsatisfiable if and only if G does not contain a perfect matching. In particular, whenever the number of underlying vertices in G is odd, a simple parity argument shows that $\neg NPM_G$ is unsatisfiable. Although parity is not expressible by formulas allowed in CP proofs (formulas that express linear inequalities), it turns out that there are still short tree-like CP proofs. The following lemma shows that $\neg NPM_G$ is unsatisfiable if and only if there is a polynomial-size tree-like CP proof of NPM_G. In particular, this lemma implies that PHP has short tree-like CP proofs.

Lemma 1: If $G = (V, E)$, $|V| = n$, is a graph which does not contain a perfect matching, then $\neg NPM_G$ has a polynomial-sized tree-like CP refutation.

Proof: We will argue informally that such proofs exist. Let $G = (V, E)$ be a graph with no perfect matching, and assume $\neg NPM_G$. By Tutte's theorem [LP, p. 84], there is a set of vertices $S \subseteq V$ so that the set of odd components of $G \setminus S$ is larger than S. The proof proceeds in two steps. Let H be an odd component of $G \setminus S$ having $2m + 1$ vertices. Label the interior edges of H with labels P_{ij}, the exterior edges (joining H to S) with labels Q_{ij}. Step one is to derive

$$\sum_{i \in v(H),\ j \in v(S)} Q_{ij} \geq 1.$$

First, derive

$$2(\sum_{i,j \in v(H)} P_{ij}) + \sum_{i \in v(H), j \in v(S)} Q_{ij} \geq 2m + 1$$

from $\neg NPM_G$. Next, derive

$$2(\sum_{i,j \in v(H)} P_{ij}) \leq 2m + 1.$$

Dividing and then multiplying again, we get:

$$2(\sum_{i,j \in v(H)} P_{ij}) \leq 2m.$$

Then using the addition rule, we obtain: $\sum_{i \in v(H),\ j \in v(S)} Q_{ij} \geq 1$, which is what we wanted. Now let k be the number of odd components of $G \setminus S$, and let l be the cardinality of S. Summing all the inequalities derived in step one, we have:

$$\sum_{i \in v(G \setminus S),\ j \in v(S)} Q_{ij} \geq k.$$

Step two is to derive the inequality

$$\sum_{i \in v(G \setminus S)} Q_{ij} \leq 1,$$

for all $j \in v(S)$. (Again, this is possible since we assumed $\neg NPM_G$.) Summing all these inequalities from step two, we get:

$$\sum_{i \in v(G \setminus S),\ j \in v(S)} Q_{ij} \leq l.$$

But this contradicts the fact that $k > l$. $\square$

Another family of tautologies based on a parity argument are Tseitin's graph tautologies. Tseitin's tautology for the graph G, $TS(G)$, states that if G is a connected graph where each vertex has an associated 0-1 value (called the *charge* of the vertex), and if the total sum of the charges is congruent to 1 mod 2, then there must exist a vertex, v, such that the degree (mod 2) of v does not equal the charge of v. If the underlying graph G has n vertices and degree k, then the associated graph tautology is expressible by a formula of size $O(2^k n)$: the underlying variables are E_{ij}, for all i, j such that e_{ij} is an edge in G, and the tautology $TS(G)$ states that there exists a vertex i such that the mod 2 sum of the variables E_{ik} is 0. The standard refutation of $\neg TS(G)$ shows that assuming $\neg TS(G)$, the mod 2 sum of all edges incident to all vertices is 1, since the number of vertices is odd. But this sum must be even since we are counting each edge exactly twice. It is not possible to formalize this particular argument in Cutting Planes since we cannot count modulo 2 with depth-1 threshold formulas. In fact, we conjecture that any CP proof requires superpolynomial size when G is an expander graph, although, so far, super-linear lower bounds have not been established. (This family of tautologies was first introduced by Tseitin [T]; Urquhart [U] showed that any Resolution proof of these tautologies requires exponential size.)

The hard example for which we are able to obtain a lower bound is based on a slight generalization of the pigeonhole principle. This example will be discussed in the following section.

3 The lower bound

We will obtain our lower bound by showing that any polynomial-sized tree-like Cutting Planes proof of an unsatisfiable formula f can be used to obtain an efficient communication protocol for a particular search problem related to f. We will then construct a tautology with the property that the underlying search problem is known to have high communication complexity. It turns out that this tautology is very natural and can be viewed as a generalization of the well-studied propositional pigeonhole principle. We will first review some definitions from communication complexity and a theorem due to Raz and Wigderson.

3.1 Communication complexity of search problems

Let R be a relation, $R \subseteq X \times Y \times Z$. A deterministic communication protocol A for R specifies the exchange of bits between two players, I and II, that receive as inputs $x \in X$ and $y \in Y$, respectively, and finally agree on a value $A(x, y)$ such that $(x, y, A(x, y)) \in R$. The deterministic communication complexity of R, $c(R)$, is the number of bits communicated between players I and II in the optimal protocol for R. In a probabilistic communication protocol, the two players may flip random bits. The probabilistic communication complexity of R, $c_\epsilon(R)$ is the number of bits communicated in the optimal protocol for F, where the probability of error on any input is at most ϵ. (When $\epsilon = 0$, $c_\epsilon(R) = c(R)$.)

We will now describe a particular search problem with high communication complexity. Let $Match(n)$ be the boolean function which takes a graph, G_n, on $n = 3m$ vertices as input, and outputs 1 if and only if G_n contains a matching on m edges. Let M_1 be the set of graphs on n vertices which contain m-matchings and no other edges. Let M_2 be the set of graphs on n vertices which have an independent set on $2m + 1$ vertices, and all other edges are present. Clearly M_1 is a subset of $Match^{-1}(1)$, and M_2 is a subset of $Match^{-1}(0)$. The communication complexity problem, $FindEdge(n)$, is: player I is given a graph $G_1 \in M_1$, and player II is given a graph $G_2 \in M_2$, and their job is to find an edge of G_1 which is not present in G_2. (Note that G_1 and G_2 can each be specified by $O(n \log n)$ bits.) Clearly such an edge exists by the pigeonhole principle. Raz and Wigderson [RW] showed that $FindEdge(n)$ has high deterministic communication complexity. Their proof can also be adapted to show that it has high probabilistic communication complexity as well.

Theorem 2 (Raz, Wigderson) $c_{1/n}(FindEdge(n)) = \Omega(n/\log n)$.

3.2 The Cutting Planes search problem

Let f be an unsatisfiable CNF formula. We will be interested in the following search problem associated with f: given a truth assignment, find a clause which is falsified. The underlying model for the computation is a threshold decision tree.

A threshold decision tree is a rooted, directed tree where vertices are labelled with threshold functions and edges are labelled with either 0 or 1. The leaves of the tree are labelled with clauses of f. A threshold decision tree computes the search problem for f in the obvious way: start at the root and evaluate the threshold function; follow the edge which is consistent with the value of the threshold function; continue until we hit a leaf and output the associated clause.

The threshold decision tree complexity of f, $D(f)$, is the minimum depth of any threshold decision tree for computing the search problem for f. The threshold decision tree complexity of f is a guideline to whether f will have efficient Cutting Planes proofs. The following lemma shows that an efficient (tree-like) Cutting Planes proof can be converted into a small-depth threshold decision tree for f. Unfortunately, there is no converse to this lemma. (Note that every CNF formula containing m clauses has a trivial $O(m)$-depth threshold decision tree—the tree obtained by evaluating the clauses one-by-one.)

Lemma 3: If $D(f)$ is greater than d, then any tree-like Cutting Planes proof of f must have size $exp(\Omega(d))$.

Proof: Assume that P is a size S tree-like Cutting Planes proof of f. We will describe a depth $O(\log S)$ threshold decision tree which computes the search problem associated with f. The proof is by induction on S. Clearly if the size is 1, then the unsatisfiable formula is a single, false threshold formula, so the lemma holds. Now assume that the size of P is S. Viewing P as a tree, there exists an intermediate threshold formula, f, in P such that the number of formulas above f are between $S/3$ and $2S/3$. Let the subtree of P with root formula f be denoted by P_1, and let the remainder of P (consisting of all formulas in $P \setminus P_1$, with f replaced by "$1 \geq 1$") be denoted by P_2. In our decision tree, we first query the formula f. If f evaluates to 0, we proceed on the subtree P_1; otherwise, we proceed on the subtree P_2. By the induction hypothesis, since both P_1 and P_2 have size at most $2S/3$, the height of the decision tree obtained will be $\log(2S/3) + 1 \leq O(\log S)$. To see that the decision tree actually computes the search function, notice that if f evaluates to false on a given truth assignment ϕ, then we proceed on the subproof, P_1. Since the proof is sound, and the root formula of P_1 is false on ϕ, this implies that one of the leaf formulas of P_1 must be falsified by ϕ. Similarly, if f evaluates to true, then we proceed on P_2. Again since the root formula is false, one of the leaf formulas of P_2 will be falsified by ϕ. $\square$

Let f be a tautology over $x_1, ..., x_n$, and let X, Y be a partition of the variables $x_1, ..., x_n$. Then the underlying search problem associated with f can be viewed as a relation $R_f \subseteq X \times Y \times Z$, where $X \times Y$ specifies a particular truth assignment, α, to the underlying variables, and Z specifies an initial clause that is falsified by α.

Lemma 4: If the search problem for f has a threshold decision tree of depth d where all threshold formulas have polynomial-sized weights, then there exists a deterministic communication complexity protocol for R_f (over any partition of the variables into two groups X and Y) where $O(d \log n)$ bits are sent.

Proof: Fix some partition, X, Y of $x_1, ..., x_n$. Let $t = \alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_n x_n \geq k$ be the first threshold function queried at the root of the threshold decision tree for f. Then t can be written as $A \geq B$, where $A = \sum_{x_j \in X} \alpha_j x_j$ and $B = k - \sum_{x_j \in Y} \alpha_j x_j$. Player I first communicates the value of A to Player II; this requires $O(\log n)$ many bits (assuming polynomial-sized weights). Player II then completes the computation of t, and sends the value to Player I. The two players then continue on the half of the decision tree which agrees with the value of t. The protocol terminates after d rounds, and each round requires $O(\log n)$ bits. $\square$

In a similar way, small-depth threshold decision trees with no restrictions on the weights imply probabilistic communication complexity protocols.

Lemma 5: If the search problem for f has a threshold decision tree of depth d, then there exists a probabilistic communication complexity protocol for R_f with $\epsilon \leq \frac{1}{n}$, where $O(d(\log n)^2)$ bits are sent.

Proof: As above, the two players will proceed in d rounds, at each step evaluating the threshold formula and proceeding on the consistent subtree of half the size. Let t be the first threshold formula at the root of the decision tree. It is a known result that without loss of generality, we can assume that any threshold formula on n variables has weights and threshold values bounded by $2^{O(n)}$. Again, we will write t as $A \geq B$, where Player I can compute A, and Player II can compute B. Because the weights are bounded by $2^{O(n)}$, both A and B are at most $2^{O(n)}$. Viewing A and B as binary strings of length $O(n)$, we will now describe an $O((\log n)^2)$-bit probabilistic protocol for determining whether $A \geq B$. The players will recursively examine segments of their strings until they find the lexicographically first bit in which they differ – this bit determines whether $A \geq B$. Let the first half of A be A_1, and the first half of B be B_1. Player I randomly selects a prime number, $p \in [1, n^3 \log n]$, and sends $(p, A_1 \bmod p)$ to player II. If $A_1 \bmod p \neq B_1 \bmod p$, then A_1 is different from B_1 with probability 1, and the players continue on the first half of their strings. Otherwise, the players assume that $A_1 = B_1$, and they continue on the second half of their strings. The players err in this latter case when $A \neq B$ but $A_1 \bmod p = B_1 \bmod p$, which happens if p divides $A_1 - B_1$. $A_1 - B_1$ has at most $O(n)$ prime divisors, and Player I is choosing from $O(n^3)$ primes therefore the probability of error is less than $1/n^2$. Thus, the entire protocol to determine if $A \geq B$ requires $O((\log n)^2)$ bits, and the probability of error is less than $1/n$. Since the total number of rounds is d, the total number of bits required is $O(d(\log n)^2)$. □

3.3 Hard examples for Cutting Planes

We will now describe the hard tautologies and then prove a near-linear lower bound on the threshold decision tree complexity of the underlying search problem. By Lemma 5, this will give us an exponential lower bound on the size of any tree-like Cutting Planes proof of the tautologies.

The tautology, $Match_n$, states: if m_1 is a perfect matching on $2m$ elements of $[1, .., 3m]$, $n = 3m$, and m_2 is an $m - 1$ subset of $[1, .., 3m]$, then there exists an edge in m_1 such that neither endpoint is in m_2. We will encode $m_1 \in M_1$ by a matrix of variables x^i_j, $1 \leq i \leq 3m$, $1 \leq j \leq m$, where each row x_j will specify the jth pair. Similarly, $m_2 \in M_2$ will be encoded by a matrix of variables y^i_j, $1 \leq i \leq 3m$, $1 \leq j \leq m - 1$, where row y_j will specify the jth element in the set. The negation of this principle, $\neg Match_n$, states that the matrices of variables x and y specify a valid $m_1 \in M_1$ and a valid $m_2 \in M_2$, and for all edges in m_1, one of the two endpoints of each edge is contained in m_2. (I.e., for all $i, j \leq 3m$, $i \neq j$, $k \leq m$, if $x^i_k = x^j_k = 1$, then there exists $l \leq m-1$ such that either $y_l^i = 1$ or y_l^j=1.) We express $\neg Match_n$ by the conjunction of the following clauses:

[1] $(\neg x_k^h \vee \neg x_k^i \vee \neg x_k^j)$, for all $1 \leq h, i, j \leq 3m$, $h \neq i \neq j$, $1 \leq k \leq m$.

[2] $(\bigvee_{j \neq i, 1 \leq j \leq 3m} \{x_k^j\})$, for all $1 \leq i \leq 3m$, $1 \leq k \leq m$.

[3] $(\neg x_k^i \vee \neg x_l^i)$, for all $1 \leq i \leq 3m$, $1 \leq k, l \leq m$, $k \neq l$.

[4] $(\neg y_k^i \vee \neg y_k^j)$, for all $1 \leq i, j \leq 3m$, $i \neq j$, $1 \leq k \leq m-1$.

[5] $(y_k^1 \vee \neg y_k^2 \vee .. \vee y_k^{3m})$, for all $1 \leq k \leq m-1$.

[6] $(\neg y_k^i \vee \neg y_l^i)$, for all $1 \leq i \leq 3m$, $1 \leq k, l \leq m-1$, $k \neq l$.

[7] $(\neg x_k^i \vee \neg x_k^j \vee y_1^i \vee .. \vee y_{m-1}^i \vee y_1^j \vee .. \vee y_{m-1}^j)$, for all $1 \leq k \leq m$, $1 \leq i, j \leq 3m$, $i \neq j$.

The clauses in 1-3 imply that the matrix x is in M_1; the clauses in 4-6 imply that the matrix y is in M_2, and the clauses in 7 imply that every edge (pair) in x has at least one endpoint in y. It is easy to check that number of clauses in $\neg Match_n$ is $O(n^4)$.

The search problem underlying $\neg Match_n$ is: given a truth assignment to the underlying variables x_j^i, y_k^i, find a clause in $\neg Match_n$ which is falsified. We will be interested in the restricted situation where we are given x, y which satisfy all clauses in 1-6 (i.e., x describes a valid element of M_1, and y describes a valid element of M_2). In this case, we want to determine i, j, k such that $x_k^i = x_k^j = 1$, and $y_l^i = y_l^j = 0$ for all $1 \leq l \leq m-1$. The associated communication complexity problem is: Player I is given $x \in M_1$, and Player II is given $y \in M_2$, and their goal is to communicate bits back and forth until they agree upon an edge which is present in x but not in y. We are now ready to state and prove the lower bound.

Theorem 6: Any tree-like Cutting Planes proof of $Match_n$ requires size $2^{\Omega(n/\log^3 n)}$.

Proof: Let P be a tree-like proof of $Match_n$ of size S. By Lemma 3, there exists a threshold decision tree of depth $O(\log S)$ for the underlying search problem. Now by Lemma 5, there exists a probabilistic protocol for the associated communication complexity problem (described above) where $O(\log S(\log n)^2)$ bits are sent, and $\epsilon \leq 1/n$. Now this protocol can be used to obtain a probabilistic protocol for $FindEdge(n)$ where $O(\log S(\log n)^2)$ bits are sent, and $\epsilon \leq 1/n$. If $\log S < n/\log^3 n$, this contradicts Theorem 2. $\Box$

4 Upper bounds

In this section, we will sketch a proof that $Match_n$ has polynomial-sized Frege proofs. Our proof reduces $\neg Match_n$ to the negation of the pigeonhole principle. Assuming $\neg Match_n$, we will construct formulas P_{ij}, $i \leq m$, $j \leq m-1$, such that each i, $1 \leq i \leq m$ is mapped to at least one j, $1 \leq j \leq m-1$, and each j is mapped onto by at most one $i \leq m$. Define $P_{ij} = \bigvee_{k \leq 3m}(x_i^k \wedge y_j^k)$. Using clauses from 7, it can be shown that for all $i \leq m$, there exists a $j \leq m-1$ such that P_{ij}. Using clauses from 1-6, it can also be shown that for every $j \leq m-1$, there is at most one $i \leq m$ such that P_{ij}. Thus, we have shown (informally) that $\neg Match_n$ reduces to the negation of the pigeonhole principle. The above argument can be formalized by polynomial-sized, bounded-depth Frege proofs and furthermore, Buss [B] has constructed polynomial-size Frege proofs for PHP; therefore $Match_n$ has polynomial-sized Frege proofs. Thus we have the following corollary to Theorem 6.

Corollary 7: Tree-like Cutting Planes cannot p-simulate Frege systems. Moreover, tree-like Cutting Planes cannot p-simulate bounded-depth Frege+PHP.

Conversely, it is easy to see that there is a short bounded-depth Frege proof of PHP_n from $Match_n$. Thus the principles $Match_n$ and PHP_n are polynomially-equivalent, with respect to bounded-depth Frege proofs. Why are Cutting Planes proofs of $Match_n$ much more difficult than proofs of PHP_n? Essentially, the above proof of $Match_n$ relies on the ability to count the number of true formulas in a set of small-depth formulas over the underlying variables. But in Cutting Planes proofs, we are restricted to counting arguments which work *directly* on the number of 1's in the underlying *variables.*

5 Conclusion and Open Problems

In this paper we prove exponential lower bounds for tree-like CP proofs. This result is significant because tree-like CP proofs appear to be quite general, and all known tautologies with efficient CP proofs also have efficient tree-like CP proofs. Our proof introduces a natural search problem which underlies a CP refutation. We obtain the CP lower bound by showing that a short CP proof for our tautologies implies an efficient solution for the underlying search problem; but this contradicts known lower bounds for the search problem.

The central open problem is to prove a superpolynomial lower bound on the size of non-tree-like CP proofs, and in particular a lower bound for Tseitin's graph tautologies. The methods of this paper will not allow us to prove bounds for these tautologies because there *do* exist small-depth threshold decision trees for the search problem which underlies the graph tautologies. Moreover, it is possible to extend the Cutting Planes system to allow polynomial-sized proofs of the graph clauses. The extension that is needed allows branching on an arbitrary threshold formula t, as well as on the negation of t, and then deriving a contradiction along each branch separately. Without this extension, it is conjectured that the graph examples do not admit polynomial-sized CP proofs. We conclude by mentioning several other open problems.

(a) Is it possible to use the techniques of this paper to show that randomly generated unsatisfiable formulas require superpolynomial-sized tree-like CP proofs, with high probability?

(b) Can tree-like CP p-simulate CP? It was shown by Krajíček that depth $d+1$ tree-like Frege systems can p-simulate depth d Frege systems, and that the converse fails to hold. Thus we might expect that tree-like CP cannot p-simulate CP, but are there potential tautologies which might witness this separation?

(c) Another important problem is to prove even an n^2 lower bound on the number of distinct subformulas in a Frege proof. One approach to solving this problem is to characterize a particular non-trivial search problem associated with a tautology and show that any short Frege proof must solve this problem. Note that the search problem introduced in this paper becomes trivial once we allow arbitrary formulas in the proof. Therefore, we would need to formulate a new search problem.

References

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

[BIK] 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.

[C1] Clote, P. "Cutting planes and constant depth Frege proofs," Manuscript 1993.

[C2] Clote, P. "On polynomial size Frege proofs of certain combinatorial principles," 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.

[G1] Goerdt, A. "Cutting plane versus Frege proof systems," *Lecture Notes in Computer Science*, 533.

[G2] Goerdt, A. "Hard examples for the cutting planes proof system with bounded degree of falsity," CSL, 1991.

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

[K] Krajíček , J. Personal communication, 1992.

[LP] Lovász, L., Plummer, M.D., *Matching Theory*, Annals of Discrete Mathematics 121, North-Holland, Amsterdam, 1986.

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

[T] Tseitin, G.S. "On the complexity of derivations in the propositional calculus," *Studies in Mathematics and Mathematical Logic, Part II*, A.O. Slisenko, ed., 1970, pp. 115-125.

[U] Urquhart, A., "Hard examples for Resolution," *Journal of ACM*, Vol. 34, Number 1, January 1987, pp. 209-219.