A lower bound for DLL algorithms for k-SAT

Pavel Pudlák * Russell Impagliazzo †

July 12, 1999

Abstract

There exist constants $c_k > 0$, $c_k \to 0$ as $k \to \infty$, such that every DLL algorithm for the satisfiability of sets of disjunctions of size $\leq k$ using m variables runs for at least $\Omega(2^{m(1-c_k)})$ steps on some inputs. Our proof is based on proving such a lower bound on the tree-like resolution proofs of generalized Tseitin tautologies.

1 Introduction

While NP-complete problems are all equivalent as far as their tractability, different problems can vary substantially as far as their susceptibility to heuristics. In particular, back-tracking methods for satisfiability have been shown experimentally to work rather well for random instances of 3-SAT, even when the distribution is chosen to be the hardest in terms of ratio of clauses to variables. The best fit curve for algorithms based on DLL procedures ([DLL62]) is roughly $2^{m/19.5}$ for random instances of 3-SAT, which means that such problems can be solved even with several hundred variables ([KS, CA96]). On the other hand, no comparable success has been exhibited for k-SAT for larger values of k ([Sel]).

This paper gives a partial explanation of why DLL heuristics are not successful for k-SAT for large values of k. We give a worst-case lower bound for the running time of DLL type heuristics for k-SAT which approaches exhaustive search as k increases. More precisely, since DLL algorithms provide tree-like resolution proofs of unsatisfiable formulas, for each k, we give a a lower bound on such proofs for a family of sets of contradictory k-clauses. Our lower bounds are of the form: $\Omega(2^{m(1-c_k)})$, where m is the total number of variables in the formula, and c_k approaches 0 as k increases. The tautologies we prove lower

*Partially supported by grant A1019901 of the Academy of Sciences of the Czech Republic and by a cooperative research grant INT-9600919/ME-103 from the NSF (USA) and the MŠMT (Czech Republic)

†Research Supported by NSF Award CCR-9734911, Sloan Research Fellowship BR-3311, by a cooperative research grant INT-9600919/ME-103 from the NSF (USA) and the MŠMT (Czech Republic), and USA-Israel BSF Grant 97-00188

bounds for are variants of the Tseitin graph tautologies for hypergraphs. It should be noted that our tautologies are systems of linear equations mod 2. By results of [BI99], it follows that, for unrestricted resolution, the tautologies have proofs of width $m/2 + o(m)$ and size $2^{m/2+o(m)}$. So our bounds cannot be derived by the methods of [BP96, BW99].

Known upper bounds for k-Satisfiability have similar forms to our lower bound. A variety of algorithms improving over exhaustive search for k-Satisfiability are presented in [MS, PPZ, PPSZ, Schi93, Schi96, Scho99]; all have worst-case time bounds of the form $2^{n(1-c_k)+o(n)}$ where c_k approaches 0. The last three all have $c_k = \Theta(1/k)$, which is the best known time complexity. In [IP99], it is shown that if any k-SAT problem requires strictly exponential time, then the constants in the exponent grow with k. However, it is an interesting open problem to give a theoretical reason why the constants in the exponents should approach 1 (exhaustive search) as k increases. We at least show that this is true for the most commonly used heuristics.

In the rest of this section we shall give a more precise statement of our result. A DLL algorithm (also known as the Davis-Putnam procedure; [DLL62]) for satisfiability of a CNF formula is an algorithm that searches systematically the tree of all assignments to the variables and checks whether all clauses are satisfied or at least one is falsified. In the first case is stops and accepts, in the second case it backtracks. The algorithm is thus specified by the rule by which the next considered variables is chosen. For practical purposes we need a sufficiently fast algorithm to implement the rule (in most cases the rule is very simple), but for proving a lower bound we do not need to restrict the computational complexity of the rule. The only thing that we use is that a DLL algorithm is allowed to backtrack only if it finds a falsified clause. This property ensures that from a given run of such an algorithm on an unsatisfiable formula we can construct a resolution proof of unsatisfiability of the formula. In this transformation the search tree becomes the skeleton of the resolution proof. (This is a well-known fact and also it is very easy.)

Thus the problem is reduced to proving a lower bound on tree-like resolution proofs of suitable tautologies. We shall use a natural generalization of Tseitin's graph tautologies to hypergraphs. While for the graph tautologies one needs only the expansion property of the underlying graph, for hypergraphs we need a more complicated combinatorial property.

Our main result is:

Theorem 1 *There exist constants $c_k > 0$, $c_k = O(1/k^{1/8})$, such that, for every k there exists a sequence of k-DNF tautologies which do not have tree-like resolution proofs shorter than $\Omega(2^{m(1-c_k)})$, where m is the number of variables. Hence, every DLL algorithm for k-SAT has worst-case time complexity $\Omega(2^{m(1-c_k)})$.*

2 A game for proving lower bounds on tree-like resolution

In this section, we give a partial characterization of formulas with large tree-like resolution proofs in terms of a 2-player game. Any small tree-like proof will give a good strategy for one of the players, so a good strategy for the other player yields a corresponding lower bound on proofs.

Let $\mathcal{C} = \{C_1, \ldots, C_m\}$ be a set of inconsistent clauses (disjunctions of literals) in the variables $x_1, \ldots, x_m$. Consider the following game between two players: In each round, Player 1 chooses a variable to be assigned a value. Then Player 2 chooses one of 0,1, or $*$. If 0 or 1 is chosen, no points are scored, the variable is set to the chosen bit, and the next round begins. If $*$ is chosen, then Player 2 scores one point, but Player 1 then can choose the value for the variable. The game ends when one of the clauses is forced to false by the assigned values, i.e., when all the literals in the clause are assigned false.

Lemma 1 *If $\mathcal{C}$ has a tree-like resolution refutation of size S, then Player 1 has a strategy where Player 2 can win at most $\lceil \log S \rceil$ points.*

Proof. Fix a proof of size S. Player 1 will maintain the following invariant after each round: If Player 2 has scored t points, then the partial assignment will falsify a clause in the proof, and the sub-tree rooted at this clause will be of size at most $S/2^t$. In particular, this will show the claimed bound on the number of points Player 2 can score.

Let C be the clause from the invariant at the previous round. If C is a leaf, then the game halts, since a clause in $\mathcal{C}$ has been falsified. Otherwise, Player 1 picks the variable resolved on to derive C from C_0 and C_1 in the proof. If Player 2 assigns the variable a value, at least one of the two clauses C_0, C_1 is falsified, and this is the new clause for the invariant. If Player 2 scores a point and lets Player 1 determine the value, Player 1 chooses a value to falsify the clause with the smaller sub-tree. This is at most half the size of the sub-tree for the current clause, so the invariant is maintained. $\square$

Corollary 1 *If Player 2 has a strategy which scores r points, then any tree-like resolution refutation of $\mathcal{C}$ has size at least 2^r.*

3 A Family of Hypergraph Tautologies

We will prove lower bounds for tree-like resolution proofs for a generalization of the Tseitin graph tautologies to hypergraphs. The Tseitin graph tautologies express that the number of odd degree nodes in a graph is even. To make this expressible as a set of small clauses, we restrict first to sub-graphs of a fixed, small degree graph G, say of degree at most k. We use one variable x_e corresponding to each edge e in G. An assignment to these variables determines a sub-graph G' whose edges are those whose corresponding variables are set to true. The nodes v of G are given a label called the charge, $c_v \in \{0,1\}$, where the sum of the c_v's is odd. We express, for each node v, the equation: $\sum_{\{e|v\in e\}} x_e \equiv c_v \ mod\ 2$ as a system of at most 2^{k-1} clauses. Since the sum of the degrees in any graph is even, and the sum of the c_v's is odd, this is contradictory. Lower bounds for resolution and restrictions of resolution for the Tseitin graph tautology for expanders G have been proved in [T68, U87, BW99].

We generalize this to hypergraphs as follows. Let $H = (V, E)$ be a hypergraph with all edges of even sizes and where each node has degree $\leq k$. That is, the edges of H are sets of vertices of even sizes and each vertex appears in $\leq k$ such sets. In order to simplify computations, we shall allow multiple edges, thus H is rather a multiset. Let $c : V \to \{0,1\}$, with $\sum_{v\in V} c_v \equiv 1 \ mod\ 2$. Since d is even, as before, the total degrees of all vertices in any subgraph of H should be even. As before, we add a variable x_e corresponding to each edge e in G. We express, for each node v, the equation: $\sum_{\{e|v\in e\}} x_e \equiv c_v \ mod\ 2$ (where the sum has one appearance of x_e for each occurrence of v in e) as a system of at most 2^{k-1} k-clauses. Since the sum of the degrees in any subgraph is even, and the sum of the c_v's is odd, the set of all these clauses is contradictory. The negation of the conjunction of these clauses is therefore a tautology, which we will refer to as $T_{H,c}$. We will assume that $|V|$ is odd and use as a default c the function which assigns all nodes value 1, and refer to the resulting hypergraph tautology as T_H. Let $n = |V|$, $m = |E|$. We will show using a probabilistic argument that there are hypergraphs H such that any tree-like resolution refutation of $\neg T_H$ requires size at least $\Omega(2^{m(1-c_k)})$, where $c_k = O(k^{-1/8})$. Since m is the number of variables, and $\neg T_H$ is a k-CNF formula, this yields the claimed lower bound for DLL for k-SAT.

4 A combinatorial condition necessary for T_H to have small proofs

In this section, we introduce a general strategy for Player 2 in the game for T_H. We show that, if this strategy fails to score the desired number of points, then H has a certain combinatorial structure. In the next section, we will show that a random H does not have this structure, and so conclude that T_H requires large proofs for most hypergraphs.

Let Q_i be the set of edge corresponding to the variables Player 1 has proposed in the first i rounds of the game, and let $E_i = E - Q_i$. For $e \in Q_i$, let β_e be the value that was assigned to e when it was proposed. Let $c_i(v) = c(v) + \sum_{\{e\in Q_i|v\in e\}} \beta_e$ be the relative charge of $v \in V$ after the first i rounds. For a subset $X \subseteq V$ we shall denote by $c_i(X) \equiv \sum_{v\in X} c_i(v) \ \ mod\ \ 2$ the *total charge* of the set X. The charge will always be computed mod 2. In particular, note that $c_0(V) = 1$ by assumption, and since assigning a value to any edge changes the relative charge of an even number of nodes, $c_i(V) = 1$ for all i. Note that, letting $H_i = (V, E_i)$, the remaining game is that for T_{H_i,c_i}.

Intuitively, Player 1 is trying to narrow down possible places where a charge constraint could be violated. If X is a subset of V, X is isolated at time i if every edge $e \in E_i$ has $|e \cap X| \equiv 0 \ mod\ 2$, since then no assignment to edges can change $c_i(X)$. (For example, in a graph, X is isolated if it is disconnected from $\overline{X}$.) In particular, if X is isolated and $c_i(X) = 1$, then any assignment to the remaining variables violates the charge constraint for some node in X. Although this may not be the only way to deduce that a violation must occur in a subset, we use the size of the smallest such subset as a measure of progress. In particular, observe that at the end of the game, Player 1 has found a violated constraint at some node v, and it is easy to see that then $\{v\}$ is isolated and $c_i(v) = 1$.

More precisely, let

$$L_i = \{X \subseteq V;\ \forall e \in E_i\ |e \cap X| \equiv 0\ mod\ 2\}$$

be the set of isolated sets, and let

$$\mu_i = \min\{|X|;\ X \in L_i, c_i(X) = 1\}$$

be the progress measure.

Note that, L_i depends only on Q_i, not the assigned value, so L_{i+1} can be computed immediately after Player 1 proposes the variables in the $i+1$'st round. We analyze the following strategy for Player 2: If $L_{i+1} = L_i$, respond with $*$, otherwise, respond with the value in $\{0, 1\}$ which minimizes μ_{i+1}.

Lemma 2 *Suppose $L_0 = \{\emptyset, V\}$. Then for any game where 2 plays the strategy above, there are sets $V = X_0, X_1, \ldots, X_t$, where $|X_{i+1}| \geq |X_i|/2$, for $i = 0, \ldots, t-1$ and $|X_t| = 1$ such that every edge which has an odd intersection with one of these sets has been used in the game. The number of points that Player 2 scores is at least the number of such edges minus the number of vertices of the hypergraph.*

Proof. The lemma uses four simple facts about L_i and μ_i:

Claim 1 L_i is a vector space over GF_2 (if we view sets as their characteristic vectors).

Proof. If $X_1, X_2 \in L_i$ then for every $e \in E_i$, $|e \cap (X_1 \oplus X_2)| \equiv (|e \cap X_1| + |e \cap X_2|) \equiv 0\ mod\ 2$.

Claim 2 $\mu_{i+1} \geq \mu_i/2$.

Proof. Let e be the edge proposed during round $i+1$. If every $X \in L_{i+1} \setminus L_i$ has size $\geq \mu_i/2$, then there is nothing to prove, since any choice of β_e satisfies the Claim. Now suppose the contrary for some $X \in L_{i+1} \setminus L_i$. Then since $|e \cap X| \equiv 1$, (otherwise X would be in L_i), it is possible to choose β_e so that

$c_{i+1}(X) \equiv 0$. We shall show that this choice ensures that $c_{i+1}(Y) \equiv 0$ for all $Y \in L_{i+1}$ such that $|Y| < \mu_i/2$. So let Y be such a set. If $Y \in L_i$, then e_i intersects it evenly, hence $c_{i+1}(Y) = c_i(Y) = 0$. It remains to consider such a Y which is not in L_i. Then its intersection with e is odd, hence

$$|e \cap (X \oplus Y)| \equiv |e \cap X| + |e \cap Y| \equiv 0 \ mod\ 2,$$

thus $X \oplus Y \in L_i$. Also $|X \oplus Y| \leq |X| + |Y| < \mu_i$, hence $c_{i+1}(X \oplus Y) = c_i(X \oplus Y) = 0$. Then we have

$$c_{i+1}(Y) = c_{i+1}(X) \oplus c_{i+1}(X \oplus Y) = c_{i+1}(X) = 0.$$

Claim 3 If $L_{i+1} = L_i$, then $\mu_{i+1} = \mu_i$.

Proof. This is also trivial, since player 1's choice of β_e does not affect the charge of any set in L_i.

Claim 4 The number of moves that do not score points is at most $|V|$.

Proof. Since $L_i \subseteq L_{i+1}$ and both are vector spaces, the dimension must increase unless they are equal, in which case a point is scored. The maximal dimension is $|V|$.

Now we can finish the proof of the lemma. Let $X_0 = V$. Let i_j be the j'th time when $L_j \neq L_{j+1}$. For $j > 0$, let X_j be an element of L_{i_j+1} of size μ_{i_j+1} such that $c_{i_j+1}(X_j) = 1$. The bound on the size of X_j follows from Claim 2. By definition, for an $X \in L_p$ all edges which have an odd intersection with X must have been used in rounds up to the p-th round. Thus all edges which have an odd intersection with some X_j must be used in the game. The bound on the number of scored points now follows from Claim 4. □

We would like to show that in a random hypergraph most of the edges have an odd intersection with at least one of the sets X_j. We shall prove it by first estimating the probability that a single edge of size d has an odd intersection with some X_j. As the sets X_j may overlap very much, in fact, consecutive sets may be equal, the corresponding probabilities for these sets may be very dependent. In order to get more independence we shall take a subsequence of this sequence and lower bound the number of edges that have an odd intersection with at least one of the sets by the corresponding quantity for the subsequence. Therefore we modify the lemma as follows.

Lemma 3 *Suppose $L_0 = \{\emptyset, V\}$. Then for any game where 2 plays the strategy above, there are subsets of V $Y_1, \ldots, Y_s$ such that*

1. $s = \frac{\log d}{4} - 2$,
2. $|\bigcup Y_i| \leq n/2$,
3. $|Y_s| \geq n/\sqrt{d}$,

4. for all $i = 1, \ldots, s-1$, $|Y_i \setminus (Y_{i+1} \cup \ldots \cup Y_s)| \geq n/\sqrt{d}$

and the number of points that Player 2 scores is at least the number of edges which have odd intersections with at least one of these sets minus the number of vertices of the hypergraph.

Proof. Given a sequence $\{X_i\}$, select a subsequence $\{Y_j\}$ as follows. Let Y_1 be the first element of $\{X_i\}$ such that $|Y_1| \leq n/4$. Assuming that first i elements have been chosen, let Y_{i+1} be the first next element in $\{X_i\}$ such that $|Y_{i+1}| \leq |Y_i|/2$. By the bound on the sizes of consecutive elements of $\{X_i\}$, we have $|Y|_1 \geq n/8$ and $|Y_{i+1}| \geq |Y_i|/4$ for $i = 2, \ldots$. Thus if we continue this process for $s = \frac{\log d}{4} - 2$ steps, every element will have size at least $n/\sqrt{d}$. Conditions 2 and 4 follow from the bound on Y_i's using the formulas for the sums of geometric progressions. □

5 Construction of the hypergraphs

Let $d > 0$ be an even integer, and set $k = d^2 + 2$.

We shall construct the hypergraph H as the union of two hypergraphs H_0 and H_1. For H_0 we simply take a path of edges of size 2 (thus H_0 is a graph) going through all vertices; so $|H_0| = n - 1$. This subgraph ensures that $L_0 = \{\emptyset, V\}$, since every nonempty proper subset of V has to have an edge of the path either entering or leaving it. With increasing d the ratio $|H_1| : n$ will go to infinity, thus H_0 will have little influence on the lower bound.

5.1 Constructing H_1

The main part of the hypergraph H will be H_1. We will construct it by choosing randomly edges of size d, for d an even number depending on k. The choice will not be completely random, as we need to ensure that the maximal degree of H_1 is $k - 2$ (recall that H_0 has maximal degree 2).

We construct H_1, a degree at most $k - 2 = d^2$ hypergraph, with even size edges of size up to d, as follows. Let $U = U_0$ be a set with $(k-2)n = d^2 n$ elements, grouped into n batches of $k - 2$, with each batch corresponding to a variable. U represents occurrences of all variables in the edges of H_1. We iterate the following $m_1 = dn/4$ times, for $j = 1$ to m_1: Pick a random $\mathbf{e}_j \subseteq U_{j-1}$ of size d , set $U_j = U_{j-1} - \mathbf{e}_j$, and let e_j be the set of variables whose batches intersect $\mathbf{e}_j$ an odd number of times. (Usually, e_j is the set of all variables with corresponding elements appear in $\mathbf{e}_j$, but in the rare event that the same variable is chosen twice, we cancel the two occurrences.) Note that we do not use up all of U, but leave most variables with fewer than $k-2$ occurrences. H_1 has edges $e_1, \ldots, e_m$.

5.2 T_H requires large proofs

We will show that, with high probability, for any sequence of sets $\mathcal{Y} = \{Y_1, \ldots, Y_s\}$ satisfying the conditions of Lemma 3, the fraction of edges of H_1 which have odd intersections with at least one of the sets tends to 1 as k, d go to infinity.

More precisely,

Lemma 4 *There is a constant d_0 so that for any $d > d_0$, for $s = 1/4 \log d - 1$ and for sufficiently large n we have the following: With probability > 0 over the choice of H_1 as above, for any sets $Y_1, \ldots, Y_s$ satisfying the conditions of Lemma 3, there are at most $16/d^{1/4} m_1$ edges whose intersections with all of the Y_i have even size.*

First, note that the number of such sequences is at most 2^{6n}, since $|Y_i| \leq n/2^i$. So we need to show that, for any fixed such sequence, the probability of it having many edges with only even intersections is less than 2^{-6n}.

Let $Y_1, \ldots, Y_s$ be a fixed sequence of sets satisfying the conditions of Lemma 3. For each $i = 1, \ldots, s$ pick a set $Z_i \subseteq Y_i - \cup_{j=i+1}^{j=s} Y_j$ with $|Z_i| = n/(d^{1/2})$ and let A_i be the corresponding $nd^{3/2}$ elements of U.

Lemma 5 *The probability that there is an i with $|A_i \cap U_{m_1}| \leq nd^{3/2}/2$ is less than $snd^2(exp(-nd^{3/2}/32))$*

Proof. Follows from Chernoff bounds. □

Lemma 6 *Let $0 \leq j < m_1$, and let $\mathbf{e}_1, \ldots, \mathbf{e}_j$ be such that $|A_i \cap U_j| \geq nd^{3/2}/2$ for each $1 \leq i \leq s$. Then the conditional probability that $|e_{j+1} \cap Y_i| \equiv 0 \ mod\ 2$ for all $i = 1, \ldots, s$ is at most $2/2^s = 4/d^{1/4}$.*

Proof. Let $A'_i = A_i \cap U_j$. Pick $B_1, \ldots, B_s$ as disjoint sets of elements from U_j corresponding to variables from $V \setminus \cup_i Y_i$ so that $|B_i| = |A'_i|$. This is possible, since the number of such elements in U is at least $1/2nd^2$, and $|U_j| \geq 3/4nd^2$. Let $F_i = A'_i \cup B_i$. View $r = \mathbf{e}_{j+1}$ as being chosen in stages: First, pick $|r \cap F_i|$ for $i = 1, , s$. Then pick all the elements in $r \setminus \cup_i F_i$ Then, one by one, for $i = s$ to $i = 1$, pick the elements in $r \cap F_i$. By the time we pick the elements from F_i, all the other elements in r corresponding to Y_i have been selected. So whether $e_{j+1} \cap Y_i$ is even or odd depends on whether $r \cap A'_i$ is even or odd. But since, picking elements from F_i, we are just as likely to select elements from B_i as A'_i, this is roughly 1/2.

More precisely:

Lemma 7 *$Prob[r \cap F_i = \emptyset] \leq exp(-d^{1/2})$.*

Proof. $|F_i| = 2|A'_i| \geq nd^{3/2}$. Picking the elements of r one by one, given that no previous element was in F_i, the probability that the next is in F_i is at least $|F_i|/|U_j| \geq d^{-1/2}$. Thus, the probability that no element of F_i is chosen is at most $(1 - d^{-1/2})^d \leq exp(-d^{1/2})$. □

Lemma 8 *Let $d \geq t > 0$. Then*

$$1/2 - 1/n \leq Prob[|r \cap A_i| \ mod\ 2 = 0| \quad |r \cap F_i| = t] \leq 1/2 + 1/n.$$

Proof. Conditioned on $|r \cap F_i| = t$, $r \cap F_i$ are t random elements. Consider the probability that $|r \cap A_i| = |r \cap A'_i|$ is even conditioned on the first $t-1$ elements. The parity changes if the last element is chosen from A'_i. The probability of this depends on the number of previous elements chosen from A'_i, and is bounded between the two extremes when only elements from A'_i have been chosen and only elements from B_i, i.e. between $(|A'_i| - t - 1)/(2|A'_i| - t - 1) = 1/2(1 - (t - 1)/(2|A'_i| - t - 1) \geq 1/2(1 - t/|A' - i|) \geq 1/2(1 - 2d/d^{3/2}n) \geq 1/2 - 1/n$ and $|A'_i|/(2|A'_i| - t - 1) = 1/2(1 + (t-1)/(2|A'_i| - t - 1)) \leq 1/2 + 1/n$. □

Combining the two lemmas in the argument above, we get, for sufficiently large d and n : $Prob[|r \cap Y_i| \equiv 0 \ mod\ 2$ for $i = 1, \ldots, s] \leq (1/2 + 1/n + \exp(-d^{1/2}))^s \leq 2^{-s} \exp(s/n + s \exp(-d^{1/2})) \leq 22^{-s} = 4/d^{1/4}$.

□

Let $E_{allevens} = \{e_j| \quad |e_j \cap Y_i| \ mod\ 2 = 0$ for all $i\}$.

We have seen that, unless an unlikely event occurs, for each j, conditioned on the previous edges, $Prob[e_j \in E_{allevens}] \leq 4/d^{1/4}$.

Lemma 9 *With probability $1 - 2^{-2d^{3/4}n}$, either $|E_{allevens}| \leq 16/d^{1/4}m_1$, or $|A_i \cap U_j| < nd^{3/4}/2$ for some i, j.*

Proof. Let J be a subset of $\{1, \ldots, m_1\}$ with $|J| = 16m_1/d^{1/4}$. It follows from Lemma 6 that the probability that $e_j \in E_{allevens}$ for all $j \in J$ and that $A_i \cap U_{j-1}$ always meets the size condition is at most $(4/d^{1/4})^{|J|}$. Then the probability that $|E_{allevens}| \geq 16/d^{1/4}$ is the same as that $J \subseteq E_{allevens}$ for some such J, which is at most

$$\binom{m_1}{16m_1/d^{1/4}} (4/d^{1/4})^{16m_1/d^{1/4}} \leq (e/4)^{16m_1/d^{1/4}} \leq 2^{-2d^{3/4}n}.$$

□

If $A_i \cap U_{m_1}$ meets the minimum size bounds, so do $A_i \cap U_j$ for all $1 \leq j \leq m_1$. Thus, the probability that $|E_{allevens}| > 16/d^{1/4}m_1$ is at most $2^{-2d^{3/4}n} + snd^2(exp(-nd^{3/2}/32))$, by Lemma 5. For sufficiently large d and sufficiently large n, each of these terms is less than $1/42^{6n}$. So there is an H_1 with $|E_{allevens}| \leq 16/d^{1/4}m_1$ for all sets $Y_1, \ldots, Y_s$ meeting the conditions. This proves Lemma 4

Then it follows that, for some sets $Y_1, \ldots, Y_s$ meeting the conditions, all the edges in $E_1 - E_{allevens}$ must be proposed at some point in the game. Therefore, Player 2 has earned at least $m_1 - 16m_1/d^{1/4} - n$ points, which is at least $m(1 - 2/d - 16/d^{1/4})$. Thus, (since $d = \Omega(k^{1/2})$) any tree-like resolution refutation of T_H requires size $2^{m(1-\Theta(k^{-1/8}))}$.

6 Conclusions and Open Problems

We show that for sufficiently large constants k, DLL algorithms are not much better than exhaustive search in the worst-case. This points to two obvious places for improvement: meaningful results for small values of k and average-case lower bounds. In [CS88, BP96, BKPS98], exponential lower bounds for random $k-CNF$'s with fixed densities are shown, both for some versions of DLL algorithms and for arbitrary resolution proofs. However, their emphasis was on proving exponential lower bounds for large densities, where the formula is overly constrained, rather than obtaining the best constants in the exponents for the hardest instances. It would also be interesting to try to improve the upper bounds known for these approaches.

References

[BKPS98] P. Beame, R. Karp, T. Pitassi, M. Saks. On the Complexity of Unsatisfiability Proofs for Random k-CNF Formulas. Submitted.

[BP96] P.Beame, T. Pitassi. Simplified and Improved resolution lower bounds. In *37th Annual Symposium on Foundations of Computer Science*, pp 274-282. Burlington, VT, October 1996. IEEE.

[BI99] E. Ben-Sasson, and R. Impagliazzo, Random CNF's are Hard for the Polynomial Calculus, manuscript, 1999.

[BW99] E. Ben-Sasson, A. Wigderson. Short Proofs are Narrow - Resolution made Simple. In *STOC99*.

[CA96] J. Crawford and L. Auton, Experimental Results on the Crossover Point in Random 3SAT, *Artificial Intelligence Journal*, 1996.

[CS88] V. Chvátal, E. Szemerédi Many Hard Examples for Resolution In *J. of the ACM*, Vol 35 No. 4, pp. 759-768.

[DLL62] M. Davis, G. Logemann, D. Loveland. A Machine program for theorem proving. In *Communications of the ACM*, 5:394-397, 1962.

[H85] A. Haken. The Intractability of Resolution. In *Theoretical Computer Science*, 39 (1985), pp. 297-308.

[IP99] R. Impagliazzo and R. Paturi, Complexity of k-SAT, *Proceedings, Computational Complexity '99.*

[KS] Kirkpatrick, Scott and Selman, Bart. Critical Behavior in the Satisfiability of Random Boolean Expressions. *Science*, Vol. 264, May 1994, 1297–1301.

[MS] Monien, B. and Speckenmeyer, E., (1985), Solving Satisfiability In Less Than 2^n Steps, Discrete Applied Mathematics **10**, pp. 287–295.

[PPZ] Paturi, R., Pudlák, P., and Zane, F., (1997), Satisfiability Coding Lemma, *in* Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science, October 1997.

[PPSZ] Paturi, R., Pudlák, P., Saks, M., and Zane F., (1998), An Improved Exponential-time Algorithm for k-SAT, *1998 Annual IEEE Symposium on Foundations of Computer Science.*

[RWY97] A. A. Razborov, A. Wigderson, A. Yao. Read-Once Branching programs, rectangular proofs of the pigeonhole principle and the transversal calculus. In Proc. of the 29th STOC, 1997, pp. 739-748.

[Schi93] Schiermeyer, I. (1993), Solving 3-Satisfiability in less than 1.579^n Steps, *in* Selected papers from CSL '92, LNCS Vol. 702, pp. 379-394.

[Schi96] Schiermeyer, I. (1996), Pure Literal Look Ahead: An $O(1.497^n)$ 3-Satisfiability Algorithm, Workshop on the Satisfiability Problem, Technical Report, Siena, April 29 - May 3, 1996; University of Köln, Report No. 96-230.

[Scho99] Schöning, U. (1999), An $O((2k/(k+1))^n * poly(n))$ algorithm for k-satisfiability, manuscript

[Sel] Selman, B., Private communication, 1999.

[T68] G.S. Tseitin On the Complexity of Derivation in Propositional Calculus. In *Studies in Constructive Mathematics and Mathematical Logic*, Part 2. Consultants Bureau, New-York-London, 1968, pp. 115-125.

[U87] A. Urquhart. Hard Examples for Resolution. In *J. of the ACM*, Vol 34 No. 1, pp. 209-219.

[U95] A. Urquhart. The Complexity of Propositional Proofs In *The Bulletin of Symbolic Logic*, Vol 1 No. 4 (1995), pp. 425-467.