Phase Transitions of PP-Complete Satisfiability Problems*

Delbert D. Bailey, Víctor Dalmau, Phokion G. Kolaitis
Computer Science Department
University of California, Santa Cruz
Santa Cruz, CA 95064, U.S.A
{dbailey,dalmau,kolaitis}@cse.ucsc.edu

Abstract

The complexity class PP consists of all decision problems solvable by polynomial-time probabilistic Turing machines. It is well known that PP is a highly intractable complexity class and that PP-complete problems are in all likelihood harder than NP-complete problems. We investigate the existence of phase transitions for a family of PP-complete Boolean satisfiability problems under the fixed clauses-to-variables ratio model. A typical member of this family is the decision problem #3SAT($\geq 2^{n/2}$): given a 3CNF-formula, is it satisfied by at least the square-root of the total number of possible truth assignments? We provide evidence to the effect that there is a critical ratio $r_{3,2}$ at which the asymptotic probability of #3SAT($\geq 2^{n/2}$) undergoes a phase transition from 1 to 0. We obtain upper and lower bounds for $r_{3,2}$ by showing that $0.9227 \leq r_{3,2} \leq 2.595$. We also carry out a set of experiments on random instances of #3SAT($\geq 2^{n/2}$) using a natural modification of the Davis-Putnam-Logemann-Loveland (DPLL) procedure. Our experimental results suggest that $r_{3,2} \approx 2.5$. Moreover, the average number of recursive calls of this modified DPLL procedure reaches a peak around 2.5 as well.

1 Introduction and Summary of Results

During the past several years, there has been an intensive investigation of random Boolean satisfiability in probability spaces parametrized by a fixed clauses-to-variables ratio. More precisely, if $k \geq 2$ is an integer, n is a positive integer and r is a positive rational such that rn is an integer, then $F_k(n,r)$ denotes the space of random kCNF-formulas with n variables $x_1, \ldots, x_n$ and rn clauses that are generated uniformly and independently by selecting k variables without replacement from the n variables and then negating each variable with probability $1/2$. Much of the work in this area is aimed at establishing or at least providing evidence for the conjecture, first articulated by [Chvátal and Reed, 1992], that a phase transition occurs in the probability $p_k(n,r)$ of a random formula in $F_k(n,r)$ being satisfiable, as $n \to \infty$. Specifically, this conjecture asserts that, for every $k \geq 2$, there is a positive real number r_k such that if $r < r_k$, then $\lim_{n\to\infty} p_k(n,r) = 1$, whereas if $r > r_k$, then $\lim_{n\to\infty} p_k(n,r) = 0$.

So far, this conjecture has been established only for $k = 2$ by showing that $r_2 = 1$ [Chvátal and Reed, 1992; Fernandez de la Vega, 1992; Goerdt, 1996]. For $k \geq 3$, upper and lower bounds for r_k have been obtained analytically and experiments have been carried out that provide evidence for the existence of r_k and estimate its actual value. For $k = 3$, in particular, it has been proved that $3.26 \leq r_3 \leq 4.596$ [Achlioptas and Sorkin, 2000; Janson *et al.*, 2000] and extensive experiments have suggested that $r_3 \approx 4.2$ [Selman *et al.*, 1996]. Moreover, the experiments reveal that the median running time of the Davis-Putnam-Logemann-Loveland (DPLL) procedure for satisfiability attains a peak around 4.2. Thus, the critical ratio at which the probability of satisfiability undergoes a phase transition coincides with the ratio at which this procedure requires maximum computational effort to decide whether a random formula is satisfiable.

Boolean satisfiability is the prototypical NP-complete problem. Since many reasoning and planning problems in artificial intelligence turn out to be complete for complexity classes beyond NP, in recent years researchers have embarked on an investigation of phase transitions for such problems. For instance, it is known that STRIPS planning is complete for the class PSPACE of all polynomial-space solvable problems [Bylander, 1994]. A probabilistic analysis of STRIPS planning and an experimental comparison of different algorithms for this problem have been carried out in [Bylander, 1996]. In addition to STRIPS planning, researchers have also investigated phase transitions for the prototypical PSPACE-complete problem QSAT, which is the problem of evaluating a given quantified Boolean formula [Cadoli *et al.*, 1997; Gent and Walsh, 1999]. Actually, this investigation has mainly focused on the restriction of QSAT to random quantified Boolean formulas with two alternations (universal-existential) of quantifiers, a restriction which forms a complete problem for the class Π_2P at the second level of the polynomial hierarchy PH. The lowest level of PH is NP, while higher levels of this hierarchy consist of all decision

*Research of the authors was partially supported by NSF Grants No. CCR-9610257, CCR-9732041, and IIS-9907419

problems (or of the complements of all decision problems) computable by nondeterministic polynomial-time Turing machines using oracles from lower levels (see [Papadimitriou, 1994] for additional information on PH and its levels). Another PSPACE-complete problem closely related to QSAT is stochastic Boolean satisfiability SSAT, which is the problem of evaluating an expression consisting of existential and ranodmized quantifiers applied to a Boolean formula. Experimental results on phase transitions for SSAT have been reported in [Littman, 1999] and [Littman *et al.*, 2001].

Between NP and PSPACE lie several other important complexity classes that contain problems of significance in artificial intelligence. Two such classes, closely related to each other and of interest to us here, are #P and PP. The class #P, introduced and first studied by [Valiant, 1979a; 1979b], consists of all functions that count the number of accepting paths of nondeterministic polynomial-time Turing machines. The prototypical #P-complete problem is #SAT, i.e., the problem of counting the number of truth assignments that satisfy a CNF-formula. It is well known that numerous $\#\mathrm{P}$-complete problems arise naturally in logic, algebra, and graph theory [Valiant, 1979a; 1979b]. Moreover, #P-complete problems are encountered in artificial intelligence; these include the problem of computing Dempster's rule for combining evidence [Orponen, 1990] and the problem of computing probabilities in Bayesian belief networks [Roth, 1996]. Recently, researchers have initiated an experimental investigation of extensions of the DPLL procedure for solving #SAT. Specifically, a procedure for solving #SAT, called Counting Davis-Putnam (CDP), was presented and experiments on random 3CNF formulas from the space $F_3(n, r)$ were carried out in [Birnbaum and Lozinskii, 1999]. The main experimental finding was that the median running time of CDP reaches its peak when $r \approx 1.2$. A different DPLL extension for solving #SAT, called Decomposing Davis-Putnam (DDP), was presented in [Bayardo and Pehoushek, 2000]; this procedure is based on recursively identifying connected components in the constraint graph associated with a CNF-formula. Additional experiments on random 3CNF-formulas from $F_3(n, r)$ were conducted and it was found out that the median running time of DDP reaches its peak when $r \approx 1.5$.

In the case of the NP-complete problems kSAT, $k \geq 3$, the peak in the median running time of the DPLL procedure occurs at the critical ratio at which the probability of satisfiability appears to undergo a phase transition. Since #SAT is a counting problem (returning numbers as answers) and not a decision problem (returning "yes" or "no" as answers), it is not meaningful to associate with it a probability of getting a "yes" answer; therefore, it does not seem possible to correlate the peak in the median running times of algorithms for #SAT with a structural phase transition of #SAT. Nonetheless, there exist decision problems that in a certain sense embody the intrinsic computational complexity of #P-complete problems. These are the problems that are complete for the class PP of all decision problems solvable using a polynomial-time *probabilistic Turing machine*, i.e., a polynomial-time nondeterministic Turing machine M that accepts a string x if and only if at least half of the computations of M on input x are accepting. The class PP was first studied by [Simon, 1975] and [Gill, 1977], where several problems were shown to be PP-complete under polynomial-time reductions. In particular, the following decision problem, also called #SAT, is PP-complete: given a CNF-formula φ and a positive integer i, does φ have at least i satisfying truth assignments? This problem constitutes the decision version of the counting problem #SAT, which justifies the innocuous overload of notation. Another canonical PP-complete problem, which is actually a special case of #SAT, is MAJORITY SAT: given a CNF-formula, is it satisfied by at least half of the possible truth assignments to its variables? In addition, several evaluation and testing problems in probabilistic planning under various domain representations have recently been shown to be PP-complete [Littman *et al.*, 1998].

It is known that the class PP contains both NP and coNP, and is contained in PSPACE (see [Papadimitriou, 1994]). Moreover, as pointed out by [Angluin, 1980], there is a tight connection between $\#\mathrm{P}$ and PP. Specifically, $\mathrm{P}^{\#\mathrm{P}} = \mathrm{P}^{\mathrm{PP}}$, which means that the class of decision problems computable in polynomial time using $\#\mathrm{P}$ oracles coincides with the class of decision problems computable in polynomial time using PP oracles. This is precisely the sense in which PP-complete problems embody the same intrinsic computational complexity as $\#\mathrm{P}$-complete problems. Moreover, PP-complete problems (and #P-complete problems) are considered to be substantially harder than NP-complete problems, since in a technical sense they dominate all problems in the polynomial hierarchy PH. Indeed, the main result in [Toda, 1989] asserts that $\mathrm{PH} \subseteq \mathrm{P}^{\mathrm{PP}} = \mathrm{P}^{\#\mathrm{P}}$. In particular, Toda's result implies that no PP-complete problem lies in PH, unless PH collapses at one of its levels, which is considered to be a highly improbable state of affairs in complexity theory.

In [Littman, 1999], initial experiments were carried out to study the median running time of an extension of the DPLL procedure on instances (φ, i) of the PP-complete decision problem #SAT in which φ was a random 3CNF-formula drawn from $F_3(n, rn)$ and $i = 2^t$, for some nonnegative integer $t \leq n$. These experiments were also reported in [Littman *et al.*, 2001], which additionally contains a discussion on possible phase transitions for the decision problem #SAT and preliminary results concerning coarse upper and lower bounds for the critical ratios at which phase transitions may occur (in these two papers #SAT is called MAJSAT). As noted earlier, the main emphasis of both [Littman, 1999] and [Littman *et al.*, 2001] is not on #SAT or on PP-complete problems, but on stochastic Boolean satisfiability SSAT, which is a PSPACE-complete problem containing #SAT as a special case.

In this paper, we embark on a systematic investigation of phase transitions for a large family of PP-complete satisfiability problems. Specifically, for every integer $k \geq 3$ and every integer $t \geq 2$, let #kSAT($\geq 2^{n/t}$) be the following decision problem: given a kCNF-formula φ with n variables, does φ have at least $2^{n/t}$ satisfying truth assignments? In particular, for $t = 2$ and for every $k \geq 3$, we have the decision problem #kSAT($\geq 2^{n/2}$): given a kCNF-formula, is it satisfied by at least the square-root of the total number of possible truth assignments? Clearly, each problem in this family is a restriction of the decision problem #SAT. Note that, while an instance of #SAT is a pair (φ, i), an instance of #kSAT($\geq 2^{n/t}$) is just a kCNF-formula φ; this makes it possible to study the behavior of random #kSAT($\geq 2^{n/t}$) in the same framework as the one used for random kSAT. One may also consider the behavior of random MAJORITY kSAT, $k \geq 3$. In Section 3.2, however, we observe that the asymptotic behavior of random MAJORITY kSAT is trivial and that, in particular, it does not undergo any phase transition. In contrast, the state of affairs for random #kSAT($\geq 2^{n/t}$) will turn out to be by far more interesting.

We first show that, for every $k \geq 3$ and every $t \geq 2$, the problem #kSAT($\geq 2^{n/t}$) is indeed PP-complete. We conjecture that each of these problems undergoes a phase transition at some critical ratio $r_{k,t}$ of clauses to variables: as $n \to \infty$, for ratios $r < r_{k,t}$, almost all formulas in $F_k(n, r)$ are "yes" instances of #kSAT($\geq 2^{n/t}$), whereas for ratios $r > r_{k,t}$, almost all formulas in $F_k(n, r)$ are "no" instances of #kSAT($\geq 2^{n/t}$). As a first step towards this conjecture, we establish analytically upper and lower bounds for $r_{k,t}$. A standard application of Markov's inequality easily yields that $\frac{(t-1)}{t}\frac{1}{(k-\lg(2^k-1))}$ is an upper bound for $r_{k,t}$ (this was also implicit in [Littman *et al.*, 2001]). Using an elementary argument and the fact that the probability of satisfiability of random 2CNF-formulas undergoes a phase transition at $r_2 = 1$, we show that $(1 - 1/t)$ is a coarse lower bound for $r_{k,t}$. In particular, these results imply that the critical ratio $r_{3,2}$ of #3SAT($\geq 2^{n/2}$) obeys the following bounds: $0.5 \leq r_{3,2} \leq 2.595$. After this, we analyze a randomized algorithm, called Extended Unit Clause (EUC), for #3SAT($\geq 2^{n/2}$) and show that it almost surely returns a "yes" answer when $r < 0.9227$; therefore, $r_{3,2} \geq 0.9227$. Although EUC is a simple heuristic, its analysis is rather complex. This analysis is carried out by adopting and extending the powerful methodology of differential equations, first used by [Achlioptas, 2000] to derive improved lower bounds for the critical ratio r_3 of random 3CNF-formulas.

Finally, we complement these analytical results with a set of experiments for #3SAT($\geq 2^{n/2}$) by implementing a modification of the Counting Davis-Putnam procedure (CDP) and running it on formulas drawn from $F_3(n, rn)$. Our experimental results suggest that the probability of #3SAT($\geq 2^{n/2}$) undergoes a phase transition when $r \approx 2.5$. Thus, the 2.595 upper bound for $r_{3,2}$ obtained using Markov's inequality turns out to be remarkably close to the value of $r_{3,2}$ suggested by the experiments. Moreover, the average number of recursive calls of the modified CDP procedure reaches a peak around the same critical ratio 2.5.

2 PP-completeness of #kSAT($\geq 2^{n/t}$)

In Valiant [Valiant, 1979a], the counting problem #SAT was shown to be #P-complete via *parsimonious* reductions, i.e., every problem in #P can be reduced to #SAT via a polynomial-time reduction that preserves the number of solutions. Moreover, the same holds true for the counting versions of many other NP-complete problems, including #kSAT, the restriction of #SAT to kCNF-formulas. We now use this fact to identify a large family of PP-complete problems.

Proposition 2.1: *For every integer $k \geq 3$ and every integer $t \geq 2$, the decision problem* #kSAT($\geq 2^{n/t}$) is PP-complete. In particular, #3SAT($\geq 2^{n/2}$) is PP-complete.

Proof: For concreteness, in what follows we show that #3SAT($\geq 2^{n/2}$) is PP-complete. Let Q be the following problem: given a 3CNF-formula ψ and a positive integer i, does ψ have at least 2^i satisfying truth assignments? Since #3SAT is a #P-complete problem under parsimonious reductions, there is a polynomial-time transformation such that, given a CNF-formula φ with variables $x_1, \ldots, x_n$, it produces a 3CNF-formula ψ whose variables include $x_1, \ldots, x_n$ and has the same number of satisfying truth assignments as φ. Consequently, φ is a "yes" instance of MAJORITY SAT (i.e., it has at least 2^{n-1} satisfying truth assignments) if and only if $(\psi, n-1)$ is a "yes" instance of Q. Consequently, Q is PP-complete.

We now show that there is a polynomial-time reduction of Q to #3SAT($\geq 2^{n/2}$). Given a 3CNF-formula ψ with variables $x_1, \ldots, x_n$ and a positive integer i, we can construct in polynomial time a 3CNF-formula χ with variables $x_1, \ldots, x_n, y_1, \ldots, y_n$ that is tautologically equivalent to the CNF-formula $\psi \wedge y_{n-i+1} \wedge \cdots \wedge y_n$. It is clear that $\#\chi = 2^{n-i}\#\psi$, where $\#\chi$ and $\#\psi$ denote the numbers of truth assignments that satisfy χ and ψ respectively. Consequently, ψ has at least 2^i satisfying truth assignments if and only if χ has at least $2^n = 2^{2n/2}$ satisying truth assignments. ∎

3 Upper and Lower Bounds for #kSAT($\geq 2^{n/t}$)

Let $X_k^{n,r}$ be the random variable on $F_k(n, r)$ such that $X_k^{n,r}(\varphi)$ is the number of truth assignments on $x_1, \ldots, x_n$ that satisfy φ, where φ is a random kCNF-formula in $F_k(n, r)$. Thus, φ is a "yes" instance of #kSAT($\geq 2^{n/t}$) if and only if $X_k^{n,r}(\varphi) \geq 2^{n/t}$. We now have all the notation in place to formulate the following conjecture for the family of problems #kSAT($\geq 2^{n/t}$), where $k \geq 3$ and $t \geq 2$.

Conjecture 3.1: For every integer $k \geq 3$ and every integer $t \geq 2$, there is a positive real number $r_{k,t}$ such that:

- If $r < r_{k,t}$, then $\lim_{n\to\infty} \Pr[X_k^{n,r} \geq 2^{n/t}] = 1$.
- If $r > r_{k,t}$, then $\lim_{n\to\infty} \Pr[X_k^{n,r} \geq 2^{n/t}] = 0$.

We have not been able to settle this conjecture, which appears to be as difficult as the conjecture concerning phase transitions of random kSAT, $k \geq 3$. In what follows, however, we establish certain analytical results that yield upper and lower bounds for the value of $r_{k,t}$; in particular, these results demonstrate that the asymptotic behavior of random #kSAT($\geq 2^{n/t}$) is non-trivial.

3.1 Upper Bounds for #kSAT($\geq 2^{n/t}$)

Let X be a random variable taking nonnegative values and having finite expectation $E(X)$. Markov's inequality is a basic result in probability theory which asserts that if s is a positive real number, then $\Pr[X \geq s] \leq \frac{E(X)}{s}$. The special case of this inequality with $s = 1$ has been used in the past to obtain a coarse upper bound for the critical ratio r_k in random kSAT. We now use the full power of Markov's inequality to obtain an upper bound for $r_{k,t}$. As usual, $\lg(x)$ denotes the logarithm of x in base 2.

Proposition 3.2: *Let $k \geq 3$ and $t \geq 2$ be two integers. For every positive rational number $r > \frac{(t-1)}{t}\frac{1}{(k-\lg(2^k-1))}$,*

$$\lim_{n\to\infty} \Pr[X_k^{n,r} \geq 2^{n/t}] = 0.$$

It follows that if $r_{k,t}$ exists, then $r_{k,t} \leq \frac{(t-1)}{t}\frac{1}{(k-\lg(2^k-1))}$. In particular, $r_{3,2} \leq \frac{1}{2}\frac{1}{(3-\lg 7)} \approx 2.595$.

Proof: For every truth assignment α on the variables $x_1, \ldots, x_n$, let I_α be the random variable on $F_k(n,r)$ such that $I_\alpha(\varphi) = 1$, if α satisfies φ, and $I_\alpha(\varphi) = 0$, otherwise. Each I_α is a Bernoulli random variable with mean $(1-1/2^k)^{rn}$. Since $X_k^{n,r} = \Sigma_\alpha I_\alpha$, the linearity of expectation implies that $E(X_k^{n,r}) = (1-1/2^k)^{rn}2^n$. By Markov's inequality, we have that

$$\Pr[X_k^{n,r} \geq 2^{n/t}] \leq (1-1/2^k)^{rn}2^{(1-1/t)n}.$$

It follows that if r is such that $(1-1/2^k)^r 2^{(1-1/t)} < 1$, then $\lim_{n\to\infty}\Pr[X_k^{n,r} \geq 2^{n/t}] = 0$. The result then is obtained by taking logarithms in base 2 in both sides of the above inequality and solving for r. ∎

Several remarks are in order now. First, note if k is kept fixed while t is allowed to vary, then the smallest upper bound is obtained when $t = 2$. Moreover, the quantity $\frac{1}{(k-\lg(2^k-1))}$ is the coarse upper bound for the critical ratio r_k for random kSAT obtained using Markov's inequality. In particular, for random 3SAT this bound is ≈ 5.91, which is twice the bound for $r_{3,2}$ given by Proposition 3.2.

Let MAJORITY kSAT be the restriction of MAJORITY SAT to kCNF-formulas, $k \geq 2$. Obviously, a formula φ in $F_k(n,r)$ is a "yes" instance of MAJORITY kSAT if and only if $X_k^{n,r}(\varphi) \geq 2^{n-1}$. Markov's inequality implies that $\Pr[X_k^{n,r} \geq 2^{n-1}] \leq 2(1-1/2^k)^{rn}$, from which it follows that $\lim_{n\to\infty}\Pr[X_k^{n,r} \geq 2^{n-1}] = 0$, for every $k \geq 2$. Thus, for every $k \geq 2$, the asymptotic behavior of random MAJORITY kSAT is trivial; in particular, MAJORITY kSAT does not undergo any phase transition.

3.2 Lower Bounds for #kSAT($\geq 2^{n/t}$)

We say that a partial truth assignment α *covers* a clause c if it satisfies at least one of the literals of c. We also say that α *covers* a CNF-formula φ with n variables if α covers every clause of φ. Perhaps the simplest sufficient condition for φ to have at least $2^{n/t}$ satisfying truth assignments is to ensure that there is a partial assignment over $\lfloor n-n/t\rfloor$ variables covering φ. The next proposition shows that if r is small enough, then this sufficient condition is almost surely true for formulas in $F_k(n,r)$, as $n\to\infty$.

Proposition 3.3: *Let $k \geq 3$ and $t \geq 2$ be two integers. If $0 < r < 1-1/t$, then, as $n\to\infty$, almost all formulas in $F_k(n,r)$ are covered by a partial truth assignment on $\lfloor n - n/t\rfloor$ variables. Consequently, if $0 < r < 1-1/t$, then*

$$\lim_{n\to\infty}\Pr[X_k^{n,r} \geq 2^{n/t}] = 1.$$

It follows that if $r_{k,t}$ exists, then $r_{k,t} \geq 1-1/t$. In particular, $r_{3,2} \geq 0.5$.

Proof: In [Chvátal and Reed, 1992; Fernandez de la Vega, 1992; Goerdt, 1996]), it was shown that if $r < 1$, then 2CNF-formulas in $F_{2,n,r}$ are satisfiable with asymptotic probability 1. Fix a ratio $r < 1-1/t$ and consider a random formula φ in $F_k(n,r)$. By removing $(k-2)$ literals at random from every clause of φ, we obtain a random 2CNF-formula φ^* which is almost surely satisfiable. Let β be a satisfying truth assignment of φ^* and let α be the partial truth assignment obtained from β by taking for each clause a literal satisfied by α. Since $r < 1-1/t$, we have that α is a truth assignment on $\lfloor n-n/t\rfloor$ variables that covers φ^*; hence, α covers φ as well. ∎

The preceding Propositions 3.2 and 3.3 imply that, unlike MAJORITY kSAT, for every $k \geq 3$ and every $t \geq 2$, the asymptotic behavior of #kSAT($\geq 2^{n/t}$) is non-trivial.

3.3 An Improved Lower Bound for #3SAT($\geq 2^{n/2}$)

In what follows, we focus on #3SAT($\geq 2^{n/2}$). So far, we have established that if $r_{3,2}$ exists, then $0.5 \leq r_{3,2} \leq 2.595$. The main result is an improved lower bound for $r_{3,2}$.

Theorem 3.4: *For every positive real number $r < 0.9227$,*

$$\lim_{n\to\infty}\Pr[X_3^{n,r} \geq 2^{n/2}] = 1.$$

It follows that if $r_{3,2}$ exists, then $r_{3,2} \geq 0.9227$.

The remainder of this section is devoted to a discussion of the methodology used and an outline of the proof of this result. We adopt an algorithmic approach, which originated in [Chao and Franco, 1986] and has turned out to be very fruitful in establishing lower bound for the critical ratio r_3 of random 3SAT(see Achlioptas:2001 for an overview). We consider a particular randomized algorithm, called Extended Unit Clause (EUC), that takes as a input a 3CNF-formula φ on n variables and attempts to construct a *small* partial assignment α covering all clauses of φ. Algorirthm EUC succeeds if the number of variables assigned by α is $\leq \lfloor n/2\rfloor$, and fails otherwise. Our goal is to show that algorithm EUC succeeds almost surely on formulas φ from $F_3(n,r)$ for each $r < 0.9227$. Consequently, $r_{3,2} \geq 0.9227$.

EUC Algorithm:
For $t := 1$ to n do
  If there are any 1-clauses, (*forced* step)
    pick a 1-clause uniformly at random and satisfy it.
  Otherwise, (*free* step)
    pick an unassigned variable uniformly at random and remove all literals involving that variable in all remaining clauses.
Return *true* if the number of assigned variables is $\leq \lfloor n/2\rfloor$;
  otherwise, return *false*.

To analyze the average performance of algorithm EUC we use the *differential equations methodology* (DEM), initially introduced in [Achlioptas, 2000] and described more extensively in [Achlioptas, 2001]. Due to space limitations, we give only a high-level description of how DEM can be applied to the analysis of algorithm EUC and also outline the steps in the derivation of the improved lower bound 0.9227.

Since DEM was developed to analyze satisfiability testing algorithms, it should not be surprising that certain modifications are needed so that it can be applied to counting algorithms, such as EUC. The main component of EUC not handled directly by DEM is the *free step*, since in a satisfiability context it is always a better strategy to assign a value to the selected variable, instead of removing all the literals involving that variable. We will describe *where* and *how* we extend DEM to handle free steps.

Let $V(t)$ be the random set of variables remaining at iteration t $(0 \leq t \leq n)$ and let $S_i(t)$ denote the set of random i-clauses $(1 \leq i \leq 3)$ remaining at iteration t. To trace the value of $|S_i(t)|$, we rely on the assumption that, at every iteration of the execution of the algorithm being considered, a property called *uniform randomness* is maintained. This property asserts that in every iteration $0 \leq t \leq n$, conditional on $|V(t)| = n'$ and $|S_i(t)| = m'$, $S_i(t)$ is drawn from $F_i(n', m')$. In [Achlioptas, 2001], a protocol, called *card game*, is presented; this protocol restricts the possible ways in which a variable can be selected and assigned. It is shown that any algorithm obeying that protocol. satisfies the uniform randomness property. Unfortunately, due to the presence of the free steps, algorithm EUC does not satisfy the card protocol, unlike the majority of algorithms for satisfiability analyzed so far. It is tedious, but straightforward, to show that the natural generalization of the card game in which we allow the elimination of the literals involving the selected variable guarantees the uniform randomness property.

We analyze the algorithm EUC by studying the evolution of the random variables that count the number of clauses in $S_i(t)$, $C_i(t) = |S_i(t)|$. We also need a random variable $F(t)$ that counts the number of variables assigned up to iteration t. We trace the evolution of $C_i(t)$ and $F(t)$ by using a result in [Wormald, 1995], which states that if a set of random variables $X_j(t), 1 \leq j \leq k$ evolving jointly with t such that (1) in each iteration t, the random variable $\triangle X_j(t) = X_j(t+1) - X_j(t)$ with high probability (w.h.p.) is very close to its expectation and (2) $X_j(t)$ evolves smoothly with t, then the entire evolution of $X_j(t)$ will remain close to its mean path, that is, the path that $X_j(t)$ would follow if $\triangle X_j(t)$ was, in each iteration, the value of its expectation. Furthermore, this mean path can be expressed as the set of solutions of a system of differential equations obtained by considering the scaled version of the space-state of the process obtained by dividing every parameter by n.

As discussed in [Achlioptas, 2001], Wormald's theorem guarantees that the value of the random variables considered differs in $o(n)$ from its mean path. This possible deviation produces some difficulties in our analysis; indeed, at each iteration we need to know the value of $C_1(t)$ with far more precision, because depending on the *exact* value taken by $C_1(t)$ algorithm EUC performs different operations. To settle this technical difficulty, Achlioptas derived an elegant solution, called the *lazy-server* lemma. Intutively, this lemma states that the aforementioned difficulty can be overcome if, instead of handling unit clauses deterministically as soon as they appear, at iteration t we take care of unit clauses with probability p and perform a free step with probability $(1-p)$, where p has to be chosen appropriately.

An additional technical difficulty remains. As discussed in [Achlioptas, 2001], condition (2) of Wormald's theorem does not hold when the iteration t is getting close to n. This second problem is fixed by determining an iteration $t^* = \lfloor (1-\epsilon)n \rfloor$ at which our algorithm will stop the iterative process and it will deal with the remaining formula in a deterministic fashion. We now modify algorithm EUC by incorporating the features described above and obtain the following algorithm:

EUC with lazy-server policy:
For $t := 1$ to t^* do
 Set $U(t) = 1$ with probability p
 1. If $U(t) = 1$
  a) If there are 1-clauses,
   pick a 1-clause uniformly at random and satisfy it.
  b) Otherwise
   pick an unset variable uniformly at random
   and assign it uniformly at random
 2. Otherwise
  Pick an unset variable uniformly at random
  and remove all literals with that underlying
  variable in all remaining clauses.
Find a minimal covering for the remaining clauses.
Return *true* if the number of assigned variables
is $\lfloor n/2 \rfloor$; otherwise, return *false*.

Next, we compute the equation that determine the expected value for the evolution of $C_2(t)$, $C_3(t)$ and $F(t)$, conditional on the history of the random variables considered up to iteration t. Let $\mathbf{H}(t)$ be a random variable representing this history (i.e., $\mathbf{H}(t) = \langle C(0), \ldots, C(t) \rangle$, where $C(t) = (C_2(t), C_3(t), F(t))$). Since $S_i(t)$ distributes as $F_i(n-t, C_i(t))$, the expected number of clauses in $C_i(t)$ containing a given literal l is equal to $iC_i(t)/2(n-t)$. Using this, we obtain the following system of equations describing the evolution of $C_2(t)$, $C_3(t)$ and $F(t)$.

$$\mathbf{E}(\triangle C_3(t)|\mathbf{H}(t)) = -\frac{3C_3(t)}{n-t}$$
$$\mathbf{E}(\triangle C_2(t)|\mathbf{H}(t)) = -\frac{2C2(t)}{n-t} + p\frac{3C_3(t)}{2(n-t)} + (1-p)\frac{3C_3(t)}{n-t}$$
$$\mathbf{E}(\triangle F(t)|\mathbf{H}(t)) = p$$

with initial conditions $C_2(0) = 0$, $C_3(0) = rn$, $F(0) = 0$.

It is time to fix the value of p. According to the lazy-server lemma any value such that

$$p > p\frac{C_2(t)}{n-t} + (1-p)\frac{2C_2(t)}{n-t}$$

would suffice. This inequality is solved by setting

$$p = (1+\theta)\frac{(2C_2(t)/(n-t))}{1 + C_2(t)/(n-t)}$$

with $\theta > 0$.

To obtain the set of differential equations associated to the process, as discussed in [Achlioptas, 2001], we consider the scaled version of the process, $x, c_2(x), c_3(x), f(x)$ obtained by dividing every parameter $t, C_2(t), C_3(t), F(t)$ by n. We obtain the following differential equations.

$$\frac{dc_3}{dx} = -\frac{3c_3(x)}{1-x}$$
$$\frac{dc_2}{dx} = -\frac{2c_2(x)}{1-x} + p(x, c_2(x))\frac{3c_3(x)}{2(1-x)} + (1-p(x, c_2(x))\frac{3c_3(x)}{1-x}$$
$$\frac{df}{dx} = p(x, c_2(x))$$

with $p(x, c_2(x)) = (1+\theta)\frac{2c_2(x)/(1-x)}{1+c_2(x)/(1-x)}$ and initial conditions $c_3(0) = r$, $c_2(0) = 0$ and $f(0) = 0$. We solve the system numerically using the utility `dsolve` of mapple (it is easy to get $c_3(x) = r(1-x)^3$ analytically) for $r = 0.9227$, $\epsilon = 10^{-2}$ and $\theta = 10^{-5}$.

Now we are almost done. First, we can see that for every $1 \le t \le t*$ w.h.p. $S_0(t) = \emptyset$ by testing numerically that $p(x, c_2(x)) < 1 - 10^{-1}$ if $x \in [0..1 - 10^{-2}]$ and appealing to the lazy-server lemma. Moreover, we get $f(1 - 10^{-2}) < 0.49978$ and, transforming this result back to the randomized space-state, we can infer that $f(t^*) < 0.49978n + o(n)$. Similarly, the number of remaining clauses at that iteration is $C_2(t^*) + C_3(t^*) = c_2(1 - 10^{-2}) + c_3(1 - 10^{-2}) + o(n) < 0.00021n + o(n)$. It is easy to verify that the ratio of clauses to variables at iteration t^* is smaller than 1, and we can apply again the argument used to obtain the first naive lower bound of $1/2$ to guarantee that there exists a covering partial assignment with size $< 0.00021n$. Thus, by adding the previous quantities, we get $0.49999n < n/2$, which means that the algorithm succeeds.

4 Experimental Results for #3SAT($\geq 2^{n/2}$)

Preliminary experiments were run for random 3CNF-formulas with 4, 8, 16 and 32 variables on a SUN Ultra 5 workstation. For each space we generated 1200 random 3CNF-formulas with sizes ranging from 1 to 160 clauses in length. Each clause was generated by randomly selecting 3 variables without replacement and then negating each of them with probability of $1/2$.

Our goal was to test the formulas for being "yes" instances of #3SAT($\geq 2^{n/2}$), i.e., for having at least as many satisfying assignments as the square-root of the total number of truth assignments. For this, we implemented a threshold DPLL algorithm by modifying the basic Counting Davis-Putnam algorithm in [Birnbaum and Lozinskii, 1999] to include tracking of lower and upper bounds on the count and early termination if the threshold is violated by the upper bound or satisfied by the lower bound.

The results are depicted in Figures 1 and 2. In both figures the horizontal axis is the ratio of the number of clauses to number of variables in the space. The ranges of formula sizes represented in the graphs are 1 to 20, 1 to 40, 1 to 80, and 1 to 160 for the 4, 8, 16 and 32 variable spaces respectively.

The phase transition graphs show for each test point the fraction of 1200 newly generated random formulas that had a number of satisfying truth assignments greater than or equal to the square-root of the total number of truth assignments. They strongly suggest that 2.5 is a critical ratio around which a phase transition occurs. The performance graphs show the average number of recursive calls required to test each formula and they exhibit a peak around the same ratio. In the test runs a range of 1 to 160 clauses was used for each space and the run-times on the SUN Ultra 5 were approximately 10, 15 and 35 minutes for the 4, 8 and 16 variable cases, and 7 hours for the 32 variable case.

After a larger set of experiments is carried out, we plan to apply finite-size scaling to further analyze the phase transition phenomenon exhibited by #3SAT($\geq 2^{n/2}$).

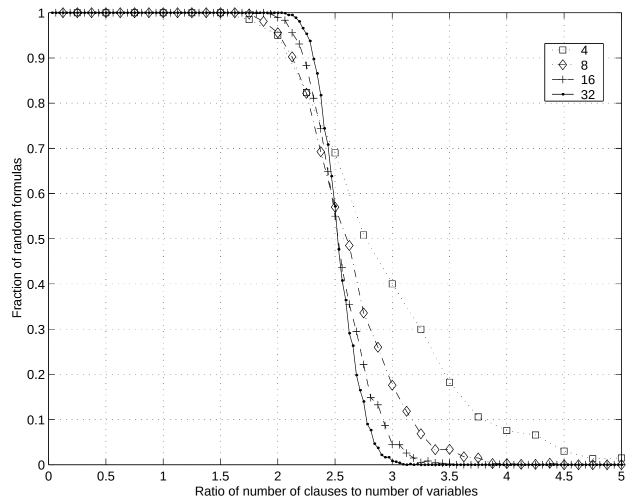

Figure 1: Phase Transition Graphs

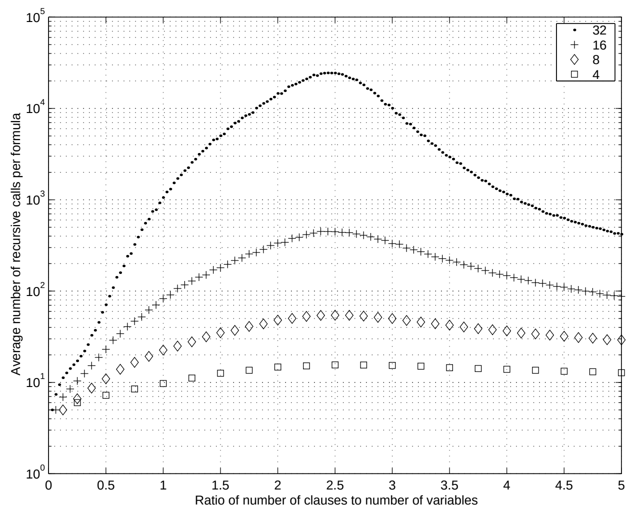

Figure 2: Performance Graphs

We conclude by pointing out that in Section 3.1.4 of [Littman *et al.*, 2001] it was suggested that for every $0 \leq \alpha \leq 1$, the critical ratio of #3SAT($\geq 2^{\alpha n}$) is given aproximately by the formula $4.2(1-\alpha)$. By taking $\alpha = 1/2$, this formula suggests that the critical ratio $r_{3,2}$ of #3SAT($\geq 2^{n/2}$) should be approximately 2.1, which is at odds with our experimental finding of 2.5 as the approximate value of $r_{3,2}$. We stand behind our experimental results; actually, we believe that this discrepancy is not caused by any significant difference in the outcome between the experiments carried out by [Littman *et al.*, 2001] and ours, but rather is due to the way in which the above formula was extrapolated from the experiments in [Littman *et al.*, 2001]. Specifically, in [Littman *et al.*, 2001] experiments were carried out by varying α and the ratio r of clauses to variables, but keeping the number of variables to a fixed value $n = 30$. The above formula $4.2(1-\alpha)$ was then derived by visual inspection of the resulting surface. We believe that, instead, the value of the critical ratio should be estimated by the crossover points of the curves obtained from experiments for different values of the number n of variables.. In any case, we see no theoretical argument or experimental evidence that a linear relationship between the critical ratio and α should hold.

Acknowledgments: We are grateful to Dimitris Achlioptas for stimulating discussions and pointers to his work on lower bounds for 3SAT via differential equations. We also wish to thank Peter Young for generously sharing with us his expertise on phase transition phenomena, and Moshe Y. Vardi for listening to an informal presentation of the results reported here and offering valuable suggestions on an earlier draft of this paper.

References

[Achlioptas and Sorkin, 2000] D. Achlioptas and G.B. Sorkin. Optimal myopic algorithms for random 3-SAT. In *41st Annual Symposium on Foundations of Computer Science*, pages 590–600, 2000.

[Achlioptas, 2000] D. Achlioptas. Setting two variables at a time yields a new lower bound for random 3-SAT. In *32nd Annual ACM Symposium on Theory of Computing*, pages 28–37, 2000.

[Achlioptas, 2001] D. Achlioptas. Lower bounds for random 3-SAT via differential equations. *Theoretical Computer Science*, 2001. To appear.

[Angluin, 1980] D. Angluin. On counting problems and the polynomial-time hierarchy. *Theoretical Computer Science*, 12:161–173, 1980.

[Bayardo and Pehoushek, 2000] R.J. Bayardo and J.D. Pehoushek. Counting models using connected components. In *7th Nat'l Conf. on Artificial Intelligence*, 2000.

[Birnbaum and Lozinskii, 1999] E. Birnbaum and E.L. Lozinskii. The good old Davis-Putnam procedure helps counting models. *Journal of Artificial Intelligence Research*, 10:457–477, 1999.

[Bylander, 1994] T. Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69:161–204, 1994.

[Bylander, 1996] T. Bylander. A probabilistic analysis of propositional STRIPS planning. *Artificial Intelligence*, 81:241–271, 1996.

[Cadoli *et al.*, 1997] M. Cadoli, A. Giovanardi, and M. Schaerf. Experimental analysis of the computational cost of evaluating quantified Boolean formulas. In *Proc. 5th Conference of the Italian Association for Artificial Intelligence*, volume 218 of *LNAI*, pages 207–218. Springer-Verlag, 1997.

[Chao and Franco, 1986] M.-T. Chao and J. Franco. Probabilistic analysis of two heuristics for the 3-satisfiability problem. *SIAM J. Comput.*, 15(4):1106–1118, 1986.

[Chvátal and Reed, 1992] V. Chvátal and B. Reed. Mick gets some (the odds are on his side). In *33th Annual Symposium on Foundations of Computer Science*, pages 620–627, 1992.

[Fernandez de la Vega, 1992] W. Fernandez de la Vega. On random 2-SAT. Manuscript, 1992.

[Gent and Walsh, 1999] I.P. Gent and T. Walsh. Beyond NP: the QSAT phase transition. In *Proc. 16th National Conference on Artificial Intelligence*, pages 653–658, 1999.

[Gill, 1977] J. Gill. Computational complexity of probabilistic Turing machines. *SIAM J. Comput.*, 6(4):675–695, 1977.

[Goerdt, 1996] A. Goerdt. A threshold for unsatisfiability. *J. Comput. System Sci.*, 53(3):469–486, 1996.

[Janson *et al.*, 2000] S. Janson, Y.C. Stamatiou, and M. Vamvakari. Bounding the unsatisfiability threshold of random 3-SAT. *Random Structures and Algorithms*, 17(2):103–116, 2000.

[Littman *et al.*, 1998] M.L. Littman, J. Goldsmith, and M. Mundhenk. The computational complexity of probabilistic planning. *Journal of Automated Reasoning*, 1998.

[Littman *et al.*, 2001] M.L. Littman, S.M. Majercik, and T. Pitassi. Stochastic Boolean satisfiability. *Journal of Automated Reasoning*, 2001. To appear.

[Littman, 1999] M. Littman. Initial experiments in stochastic satisfiability. In *Proc. of the 16th National Conference on Artificial Intelligence*, pages 667–672, 1999.

[Orponen, 1990] P. Orponen. Dempster's rule of combination is #-P-complete. *Artificial Intelligence*, 44:245–253, 1990.

[Papadimitriou, 1994] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

[Roth, 1996] D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2):273–302, 1996.

[Selman *et al.*, 1996] B. Selman, D. G. Mitchell, and H. J. Levesque. Generating hard satisfiability problems. *Artificial Intelligence*, 81(1-2):17–29, 1996.

[Simon, 1975] J. Simon. *On some central problems in computational complexity*. PhD thesis, Cornell University, Computer Science Department, 1975.

[Toda, 1989] S. Toda. On the computational power of PP and $\oplus$P. In *Proceedings 30th IEEE Symposium on Foundations of Computer Science (FOCS'89), Research Triangle Park (North Carolina, USA)*, pages 514–519, 1989.

[Valiant, 1979a] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[Valiant, 1979b] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[Wormald, 1995] N. C. Wormald. Differential equations for random processes and random graphs. *Ann. Appl. Probab.*, 5(4):1217–1235, 1995.