

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

PHASE TRANSITIONS OF BOOLEAN SATISFIABILITY VARIANTS

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Delbert D. Bailey

June 2004

The Dissertation of Delbert D. Bailey
is approved:

Professor Phokion G. Kolaitis, Chair

Professor Ira Pohl

Professor Weixiong Zhang

Robert C. Miller
Vice Chancellor for Research and
Dean of Graduate Studies

Copyright © by

Delbert D. Bailey

2004

Contents

List of Figures	v
List of Tables	vii
Abstract	viii
Dedication	ix
Acknowledgements	x
I Introduction	1
Chapter 1 Introduction	2
II Background	8
Chapter 2 Background	9
2.1 Boolean Formulas	9
2.2 k -SAT Phase Transition	11
2.3 Random Structures and Threshold Phenomena	19
2.3.1 Random Sets and Random Graphs	19
2.3.2 Threshold Functions	25
2.3.3 Coarse and Sharp Thresholds	29
2.4 General Experimental Methodology for this Research	32
2.5 Decision Versions of Optimization and Counting Problems	33
III Threshold-Counting Satisfiability Problems	35
Chapter 3 Threshold-Counting Satisfiability Problems	36
3.1 Introduction	36
3.2 Varieties of Threshold Counting Problems	40
3.3 Mean Free Field Model	41
3.4 Constant Threshold	43
3.5 Fractional Threshold	45

3.6	Exponential Threshold	47
3.6.1	Upper Bound	50
3.6.2	Lower Bound	53
3.6.3	Clustering of Number of Satisfying Assignments	54
3.6.4	Tightness of Upper Bound	55
3.7	Experiments	57
3.7.1	Method	57
3.7.2	Algorithms	58
3.7.3	Results	61
IV Bounded Satisfiability Problems		73
Chapter 4 Bounded Satisfiability		74
4.1	Introduction	74
4.2	Bounded Satisfiability Problems	77
4.3	Decision Versions of MAXSAT	78
4.3.1	Goal: at least g clauses satisfied	79
4.3.2	Bound: at most b clauses violated	79
4.4	NP-completeness of k -SAT(B)	80
4.5	Upper and Lower Bounds for k -SAT(b)	81
4.6	2BDSAT(b) Phase Transition Same as 2SAT	84
4.7	Experiments	88
4.7.1	Method	88
4.7.2	Algorithms	88
4.7.3	Results	89
4.7.4	Finite-size Scaling	103
V Generalized Satisfiability Problems		112
Chapter 5 Generalized Satisfiability Problems		113
5.1	Introduction	113
5.2	An Upper Bound for Phase Transitions in Random Monotone k -CNF Formulas	115
5.2.1	Experiments for MONOTONE-3-SAT	117
5.3	Generalized Satisfiability	122
5.3.1	Upper bounds for Generalized Satisfiability	123
5.4	1-IN-3-SAT Bound and Experiments	126
5.4.1	Upper Bound	126
5.4.2	Experiments	127
5.5	Positive 1-IN-3-SAT Bound and Experiments	130
5.5.1	Upper Bound	130
5.5.2	Experiments	132
VI Summary and Conclusions		138
Chapter 6 Summary and Conclusions		139
Bibliography		143

List of Figures

2.1	Formula Sample Spaces	12
2.2	Probability curves for 3-SAT for $n = 20, 30, 40$ variables	14
2.3	Performance curves for 3-SAT for $n = 20, 30, 40$ variables	15
3.1	Threshold Less Than	56
3.2	Threshold More Than	56
3.3	Average Search Cost Graphs for 10 and 20 Variables	61
3.4	Average Search Cost Graphs for 30 and 40 Variables	62
3.5	Probability Phase Transition Graphs	63
3.6	Peak Cost Outside Transition Window	66
3.7	Peak Cost Inside Transition Window	67
3.8	CDP vs. Basic TCDP Search Costs	71
3.9	Mean and Median Track	72
4.1	The probability curves for bounded satisfiability with $b = 0$	90
4.2	Performance behavior of the DPLL procedure for 3SAT.	91
4.3	Probability curves for 3SAT(1).	92
4.4	Performance curves for 3SAT(1)	93
4.5	Probability curves for 3SAT(2).	94
4.6	Performance curves for 3SAT(2)	95
4.7	Probability curves for 3SAT(3).	96
4.8	Performance curves for 3SAT(3)	97
4.9	Probability curves for 3SAT(4).	98
4.10	Performance curves for 3SAT(4)	99
4.11	Probability curves for 3SAT(5).	100
4.12	Performance curves for 3SAT(5).	101
4.13	Apparently separate “phase transitions” for different values of b	102
4.14	Finite-size scaling for 3SAT(0)	105
4.15	Finite-size scaling for 3SAT(1)	106
4.16	Finite-size scaling for 3SAT(2)	107
4.17	Finite-size scaling for 3SAT(3)	108
4.18	Finite-size scaling for 3SAT(4)	109
4.19	Finite-size scaling for 3SAT(5)	110
5.1	Probability curves for MONOTONE-3-SAT.	118
5.2	Probability curves for 3-SAT.	119

5.3	Performance curves for MONOTONE-3-SAT.	120
5.4	Finite-size scaling for MONOTONE-3-SAT.	121
5.5	Probability curves for 1-IN-3-SAT.	128
5.6	Finite-size scaling for 1-IN-3-SAT.	129
5.7	Variation in number of satisfied clauses with number of true variables.	131
5.8	Probability curves for POS-1-IN-3-SAT.	133
5.9	Performance curves for POS-1-IN-3-SAT.	134
5.10	Finite-size scaling for POS-1-IN-3-SAT.	135

List of Tables

3.1	Table of Truth Tables.	51
3.2	Markov Upper Bound vs. Estimate for r^*	64
4.1	Finite-size Scaling Results	104
5.1	Finite-size Scaling Fit Results for MONO-3-SAT	119
5.2	Finite-size Scaling Fit Results for 1-IN-3-SAT	129
5.3	Finite-size Scaling Fit Results for POS-1-IN-3-SAT	136

Abstract

Phase Transitions of Boolean Satisfiability Variants

by

Delbert D. Bailey

During the past decade, there has been an intensive investigation of phase transitions in Boolean satisfiability problems. This work has been motivated in large part by the apparent striking relationship of algorithm performance to the location of the phase transition. We explore three different types of generalizations of Boolean satisfiability with the goal to extend the understanding of how the structural phase transition emerges and how its emergence affects the behavior of algorithms. The generalizations consist of introducing variations in the definition of the SAT decision problem in three different ways. Threshold Counting SAT is a decision problem that allows specification of a threshold value for the number of satisfying assignments that must exist. Bounded SAT allows specification of a bound for how many clauses may be violated. Generalized SAT allows specification of the kinds of clause types that are allowed and provides for specification of more than just simple disjunctions. We explore these problems both analytically and experimentally. With the threshold problems, we discover an interesting PP-complete problem phase transition, and develop a parameterized family of decision problems that gives insight into the relationship of the performance factors of the decision problems and their associated function problems. We discover that peak algorithmic cost is not necessarily at the phase transition and also not necessarily at the cross-over point. With the Bounded SAT problems, we discover a new type of emergence for phase transitions. With the Generalized SAT problems we develop a uniform approach for determining upper bounds that yields tighter bounds than the previously reported uniform technique.

To my wife,

Donna G. Crane-Bailey.

Acknowledgements

I want to thank Phokion G.Kolaitis, my advisor, for his support, encouragement and inspiration; Victor Dalmau and Albert Atserias for insightful discussions, suggestions and generous help; Nadia Creignou and Weixiong Zhang for generously sharing and discussing the results of their research. And many others for freely giving useful advice and ideas, Jonathan Panttaja, Gene Paul, Larry Stockmeyer, Allen Van Gelder, Joel Yellin and Toby Walsh.

The work in parts III and IV of this dissertation has been previously reported in the following publications: D. Bailey, V. Dalmau and P. Kolaitis, Phase transitions of PP-complete satisfiability problems, Proceedings of the 17th International Joint Conference on Artificial Intelligence, IJCAI01, pp 183–189; D. Bailey, V. Dalmau and P. Kolaitis, Comparing phase transitions and peak cost in PP-complete satisfiability problems, Proceedings of the 18th National Conference on Artificial Intelligence, AAAI02, pp 620–626; and D. Bailey and P. Kolaitis, Phase transitions of bounded satisfiability problems, Proceedings of the 18th International Joint Conference on Artificial Intelligence, IJCAI03, pp 1187–1193.

Part I

Introduction

Chapter 1

Introduction

The concept of a phase transition has been in physics for many years, where water for example is described as existing in three different phases: solid, liquid and gas. When the temperature is varied, it moves from one phase to another at certain critical temperatures. At standard pressure, below 32 degrees Fahrenheit it is a solid and above 32 degrees it is a liquid. Moreover, there is a dramatic difference in the physical properties of these two phases. Many other examples of this type of behavior have been studied in the last quarter century, including: transitions with respect to superconductivity, superfluidity, and magnetic ordering. Physicists have developed theories to explain these phenomena based on statistical models (in other words, probabilistic models) and because of the complexity involved have found it necessary to use computer simulations to study the models, except in very special simple cases where closed form analysis can be applied [Mou84]. The idea of a phase transition can be naturally generalized with respect to probabilistic models. An observed output variable of a probabilistic model may be said to undergo a phase transition if large sudden changes of it are associated with small changes in some controlled input parameter, particularly if the output variable goes from one extreme value to another when the control parameter passes through some critical value.

With respect to physics there are two essentially different types of phase transitions. Those

that occur in nature and those that are exhibited in the probabilistic models derived from theory. On a pragmatic level the goal is to construct theories which lead to models that behave close enough to nature to allow useful predictions. On a philosophical level one can imagine such models explain nature. Classically, experiments have been used to both get ideas for theories and to test the predictions of theories. In testing, the experiments are physical, involving the manipulation and measurement of physical entities. The predictions on the other hand are analytical results derived from theory. In the case of probabilistic models in statistical mechanics the structures are so complex that analytical approaches are extremely difficult, if not impossible. It has been found that another type of experimentation can be used to make the theoretical predictions, computer simulation experiments. In 1949 Metropolis and Ulam [MU49] introduced the concept that you can learn about the parameters of a probabilistic model to an arbitrary degree of accuracy by sampling from it. The computer simulation approach treats the probabilistic model as an object which in and of itself can be experimentally investigated.

In combinatorial mathematics there is also a history of study of probabilistic models, particularly random graphs. Phase transitions have been discovered in many of these structures. For example, Erdős and Rényi [ER59, ER60] discovered that graphs with n vertices and m randomly generated edges exhibit remarkable behavior. When $\alpha = m/n$ is less than $1/2$, many small isolated clusters of connected vertices are created, but when α is greater than $1/2$, a single giant cluster is created which approaches the size of n . Analytical techniques succeeded in establishing this exact result and have yielded exact results in other simple structures. With more complex structures it is frequently the case that analysis is difficult and at best gives bounds on parameters of interest, but exact results remain open problems.

During the past decade, computer scientists have carried out an intensive study of phase transitions of algorithmic problems, particularly the NP-complete Boolean satisfiability decision problems and more recently, of decision problems that are complete for higher computational complexity classes. These investigations have shed new light on the “structure” of presumably intractable decision prob-

lems by examining them from a perspective that had been hitherto unexplored in computer science. Previously, much of the theoretical progress with respect to the complexity of problems has dealt with determining computational performance bounds for worst case input situations. This is certainly good information for an algorithm designer to know before embarking on a project. However, if the worst case inputs can be expected to never occur, or rarely occur, then worst case performance bounds lose some relevance. It becomes particularly useful to understand what the typical performance will be.

Boolean satisfiability problems are pervasive in and fundamental to computer science. Boolean satisfiability is the prototypical NP-complete problem. It was the first problem to be proven NP-complete [Coo71] and myriad problems which otherwise appear quite diverse, admit of representations involving Boolean satisfiability [GJ79]. Goldberg [Gol79], reported the surprising result that for a variety of problem instance distributions the expected time complexity of the Davis-Putnam procedure for Boolean satisfiability is polynomial. Franco and Paul [FP83] later pointed out that the distributions considered by Goldberg were unreasonable in that they yield instances which are almost surely satisfiable and can with high probability be solved in constant time. They proposed a more “reasonable” distribution model called the fixed clause-to-variable ratio, which yields instance distributions whose probable satisfiability can be controlled by a parameter, namely the ratio. Their model is not derived from consideration of typical applications. It was designed to have the characteristic that the density of satisfiable instances could be controlled by a simple parameter which makes it particularly useful for research and it has become the standard model for investigations of satisfiability. Cheeseman, Kanefsky and Taylor [CKT91] subsequently noted that several NP-complete problems have phase transitions and formulated the conjecture that all NP-complete problems exhibit phase transitions and P problems do not. This conjecture, although easily refutable, caught the attention of the research community. There have been continuing investigations of the relationship of phase transitions to complexity, motivated by the possibility that studying the phase transitions of difficult problems may ultimately lead to an understanding of the average-case complexity of the problems themselves. Chvátal and Reed [CR92a]

analytically established bounds for the k -SAT phase transition and formulated the fundamental phase transition conjecture, namely that for each k there is a corresponding critical ratio value such that random formulas with ratios less than the critical value tend to be almost surely satisfiable and formulas with ratios greater than the critical value tend to be almost surely not satisfiable. They proved that this is true for 2-SAT with a critical ratio of 1, thus refuting the Cheeseman et al. conjecture. Mitchell, Selman and Levesque [MSL92] showed experimentally, on the fixed ratio model, that 3-SAT does appear to exhibit a phase transition and, moreover, that the algorithmic average cost for the DPLL peaks near the transition. Kirkpatrick and Selman [KS94a] conducted extensive experiments for k -SAT ($2 \leq k \leq 6$) and applied the statistical mechanics technique of finite scaling to characterize and quantify the corresponding phase transitions. To date, no one has been able to analytically establish a critical ratio for k -SAT for any $k > 2$; however, there has been extensive work on analytically bounding the transition region [CF86, CF90, BFU93, FS96, Ach00, FP83, MdlV95, KMPS95, KKKS98, JSV00, DBM00], the current narrowest bounds for 3-SAT are greater than 3.42 and less than 4.506. A large variety of techniques have been used to obtain bounds. They have been increasingly more powerful, increasingly more complex and have produced increasingly tighter results, but so far this direction of investigation has not led to any new insights into the nature of the phase transition and its relation to algorithmic costs. Statistical physicists using replica symmetry breaking and the cavity method have conjectured very precise values for the 3-SAT critical ratio, 4.26675 ± 0.00015 [MMZ03]. However these techniques are not considered mathematically rigorous [AM02]. Friedgut [Fri99], using techniques developed in the study of random graphs, proved analytically that the transition for k -SAT for any $k \geq 2$, must be sharp in the sense that the region between almost surely satisfiable and almost surely not satisfiable becomes vanishingly small with increasing n . This proof yields the most powerful analytical result to date on the nature of the k -SAT transition. It does not, however, completely resolve the phase transition conjecture, since it does not establish that the transition in the limit occurs at some fixed ratio.

This dissertation has six parts. This introduction is part I. Following it, in part II, we introduce the background concepts for phase transitions in Boolean satisfiability, the k -SAT phase transition and our experimental approach. Then we report on our study of the phase transitions of some important variations of the satisfiability problem. We take a primarily experimental approach with the goal of extending the understanding of how the structural phase transition emerges and how its emergence affects the behavior of algorithms. Toward this goal, in parts III, IV and V, we explore three different types of generalizations of satisfiability. These generalizations introduce variants of the SAT decision problem from three different perspectives: counting thresholds, constraint relaxation and constraint generalization. Each of the parts begins with introductory comments specific to its area and includes discussion of complexity, analytical bounds, experimental method, algorithms, and experimental results.

Specifically, in part III, we investigate Threshold Counting SAT, a decision problem that allows specification of a threshold value for the number of satisfying assignments that must exist. First we explore the effects of various kinds of threshold functions by using the Mean Free Field model, a simplification that assumes there is no interaction between the clauses of a formula. From this we find that an exponential threshold function is the most likely to produce interesting threshold behavior. Using it, we introduce a family of PP-complete decision problems that exhibit threshold behavior on the same scale as SAT and experimentally exhibit similar phase transitions. These are the #THRESHOLD- k -SAT($\geq 2^{\alpha n}$) decision problems, where n is the number of variables and, $0 \leq \alpha < 1$, is a real. If $\alpha = 0$, this is simply the k -SAT problem. For values of α greater than 0, these decision problems experimentally show phase transition behavior similar to k -SAT but with the location of the critical ratio occurring at progressively smaller values as α approaches 1. We analytically show upper and lower bounds for these supposed phase transitions and show that the assumption, that a phase transition exists for each member of the family, implies that for each ratio of clauses-to-variables, the number of satisfying assignments for random formulas will cluster around a value that is a monotonic decreasing function of the ratio. We find that the upper bounds for the transitions in the family that the

first moment technique produces are much tighter than those it yields for 3-SAT and we analyze why this is the case. Experimentally exploring the behavior of the family with respect to the location of the phase transition and the algorithmic cost, we find that there is a tight relationship between the threshold decision problem and its associated counting problem that leads to the discovery that the location of the transition behavior does not always correspond to the location of peak algorithmic cost.

In part IV, we investigate Bounded SAT, $k\text{SAT}(b)$, a decision problem that allows specification of a bound, b , for how many clauses (constraints) may be violated. We discover a new type of emergence of the phase transition which clearly shows that the peak cost point is not tied to the phase transition location but to the cross-over point. (Actually, the relation to the cross-over point was first pointed out in 1992 [MSL92]. Evidently since the cross-over points for the k -SAT problem are close to the phase transition, many refer to the peak as being related to the phase transition location). Previously reported experiments for this problem suggested that there are separate phase transition locations for each value of the bound, b . Our experimental results show that asymptotically the locations of the transitions for different bounds appear to move toward the location of the transition for the unbounded problem. Analytically, we show that they have the same first moment upper bound as the unbounded problem regardless of the value of the bound, and in the case of $2\text{SAT}(b)$, we show the bounded versions have exactly the same phase transition location.

In part V, we investigate Schaefer Generalized SAT, a decision problem that allows specification of clause types to include more than just simple disjunctions. We find that $\text{MONOTONE-}k\text{-SAT}$ appears to have a phase transition very near, or at the same point as, k -SAT and furthermore, that the analytical first moment upper bounds for the two problems are the same. We develop a uniform method for determining the upper bounds for phase transitions in random generalized SAT problems that produces much tighter bounds than the previously published uniform technique.

Finally in part VI we summarize our findings and discuss future directions for further investigation.

Part II

Background

Chapter 2

Background

2.1 Boolean Formulas

A set of variables whose domain of values has a cardinality of 2 is called *Boolean variables*. Typically the domain is defined as either $\{0, 1\}$ or $\{\mathbf{T}, \mathbf{F}\}$. Assume $\{\mathbf{T}, \mathbf{F}\}$, then a function from $\{\mathbf{T}, \mathbf{F}\}^k$ to $\{\mathbf{T}, \mathbf{F}\}$, where k is a positive integer is called a *k-ary Boolean function*. A common denotative way to define a *k-ary Boolean function* is a *truth table*, a list all of the possible tuples of the domain along with the corresponding value of the function. *Negation* is the unary Boolean function defined to have a value of \mathbf{T} , if its input has a value of \mathbf{F} and to otherwise have a value of \mathbf{F} . The *k-ary disjunction* is the Boolean function defined to have a value of \mathbf{T} if *at least one* of the elements of the input tuple have a value of \mathbf{T} and to otherwise have a value of \mathbf{F} . The *k-ary conjunction* is the Boolean function defined to have a value of \mathbf{T} if *all* of the elements of the input tuple have a value of \mathbf{T} and to otherwise have a value of \mathbf{F} . The *Boolean connectives*, \neg , \vee , and \wedge are used to respectively represent, negation, binary disjunction and binary conjunction. A *Boolean formula* is inductively defined to be a string of Boolean variables, connectives and parentheses that can be constructed using the following rules:

1. A Boolean variable is a Boolean formula.
2. The constants **T** and **F** are Boolean formulas.
3. If φ is a Boolean formula then $(\neg\varphi)$ is a Boolean formula.
4. If φ_1 and φ_2 are Boolean formulas then $(\varphi_1 \vee \varphi_2)$ and $(\varphi_1 \wedge \varphi_2)$ are Boolean formulas.

A *literal* is a variable or the negation of a variable. A *disjunctive clause* is a disjunction of one or more literals. A *conjunctive clause* is a conjunction of one or more literals. A *conjunctive normal form* formula, CNF, is a Boolean formula that is the conjunction of *disjunctive clauses*. A *disjunctive normal form* formula, DNF, is a Boolean formula that is the disjunction of *conjunctive clauses*. *k*-CNFs and *k*-DNFs are, conjunctive or disjunctive, normal form formulas made up of only *k*-literal clauses.

With the semantics for the connectives a Boolean formula can be evaluated and accordingly used to define a Boolean function. Note that for any particular assignment of values to variables, the conjunction of all the literals which are **T** for the assignment will be **T**, and the disjunction of all the literals that are **F** for the assignment will be false. So, it is possible to express a truth table with a *k*-DNF that expresses that at least one of the **T** assignments is **T**, and also possible to express it with a *k*-CNF that expresses that all of the assignments corresponding to **F** are **F**. The most fundamental question one can ask about a Boolean formula is: is it possible to assign values to its variables that will make the formula evaluate to *true*? A formula is said to be *satisfiable* if such an assignment exists.

Many problems in AI involve determining the satisfiability of CNF Boolean formulas. The applications are broad and diverse, for example, including automatic theorem proving, scheduling, and cryptography. This has led both to the pursuit of ways to solve the satisfiability problem and ways to generate hard CNF formulas for use in the development and testing of satisfiability algorithms.

Of course the best formula test-sets for a specific application would be actual instances. For general purpose algorithm development it is more useful to have parameterized distributions of test cases. That allows easy control of the characteristics of the formulas by changing the parameters.

Usually distributions of CNF formulas are used since they are easy to randomly generate and since all propositional formulas have CNF logically equivalent formulas.

2.2 k -SAT Phase Transition

The family of Boolean satisfiability problems k -SAT, with k an integer greater than one, constitute the most thoroughly investigated collection of decision problems from the perspective of phase transitions. An instance of k -SAT is a k -CNF formula; the phase transition “control parameter” of such a formula is the ratio of the number of clauses over the number of variables occurring in the formula. Intuitively, if the number of clauses of a k -CNF formula is “much larger” than the number of its variables, then the formula is overconstrained and, thus, it is likely to be unsatisfiable. On the other hand, if the number of clauses is not “much larger” than the number of variables, then the formula is underconstrained and, thus, it is likely to be satisfiable. “Likely” is quantified for given k , number of variables, and number of clauses, by considering the fraction of all possible such formulas that are satisfiable. This fraction alternatively can be described as the probability that such a formula is satisfiable assuming all such formulas are equally likely. 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 $F_k(n, rn)$ denotes the space of random k -CNF-formulas with n variables x_1, \dots, 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 $1/2$. Figure 2.1 represents these spaces as dots on a grid. The vertical axis is the number of variables, n , and the horizontal axis is the number of clauses, m . Franco and Paul [FP83] were the first to focus on this fixed clauses-to-variables ratio model and to initiate a study of the asymptotic behavior of the probability $P_k(n, rn)$ that a random formula in $F_k(n, rn)$ is satisfiable. For each rational, r , it is possible to draw a radial line in the graph of figure 2.1 that intersects all the spaces corresponding to a fixed ratio of clauses-to-variables. The asymptotic behavior of interest is the how

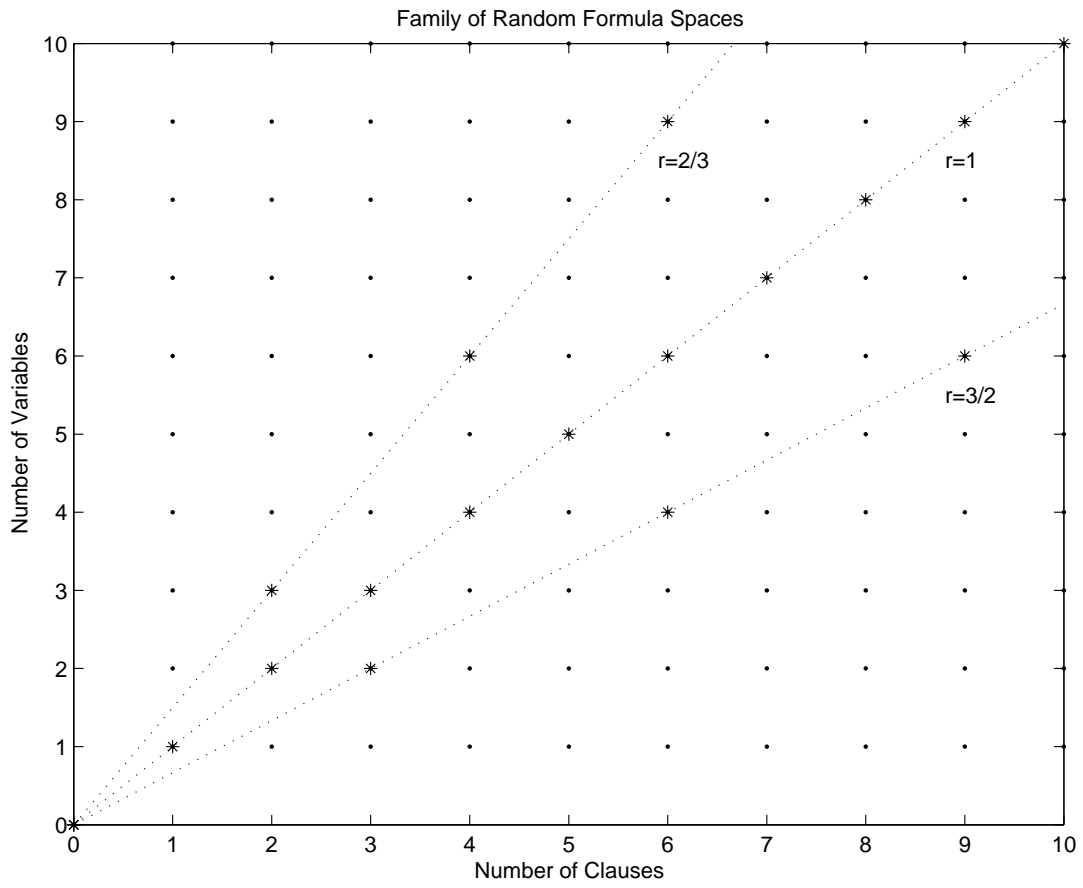


Figure 2.1: Formula Sample Spaces

the probability of satisfiability for spaces along a radial varies with movement away from the origin. Note that movement up along an ordinate has probability that must approach one and movement to the right along an abscissa has probability that must approach zero. During the past decade, much of the work in this area has been motivated from the conjecture of Chvátal and Reed [CR92a] to the effect that,

Conjecture 2.2.1 (Fundamental Conjecture) *For every $k \geq 2$, there is a positive real number r_k such that*

- *If $r < r_k$, then $\lim_{n \rightarrow \infty} P_k(n, rn) = 1$.*
- *If $r > r_k$, then $\lim_{n \rightarrow \infty} P_k(n, rn) = 0$.*

When such a *critical ratio* exists we say that a *phase transition* occurs.

The k SAT Phase Transition Conjecture, in terms of figure 2.1, says that some real ratio exists such that if a radial is drawn with this value, it will divide the grid of formula space into two distinct regions, namely, those above it and those below it. Furthermore, that the limiting values of the probabilities moving out along any of those radials in the upper region all go to one and those in the lower region all go to zero. For $k = 2$ this conjecture has been proven to be $r_2 = 1$ [CR92a, Fer92a, Goe96a]. In spite of intensive efforts by several researchers, this conjecture has not been settled thus far for $k \geq 3$. Nonetheless, progress towards establishing this conjecture has been made on two different fronts. On the experimental front, large-scale experiments with random Boolean formulas have provided evidence for the existence of a critical ratio r_k and have yielded estimates of its actual value. In particular, experiments by Selman, Mitchell and Levesque [SML96] with random 3CNF-formulas and analysis of these experiments by Kirkpatrick and Selman [KS94b] indicate that r_3 is about 4.2. On the analytical front, there has been continuous progress towards establishing progressively tighter upper and lower bounds for the value of r_3 . The best analytical results obtained to date assert that if r_3 exists, then $3.42 < r_3 < 4.507$ [KKL02, DBM00]. The experiments carried out by

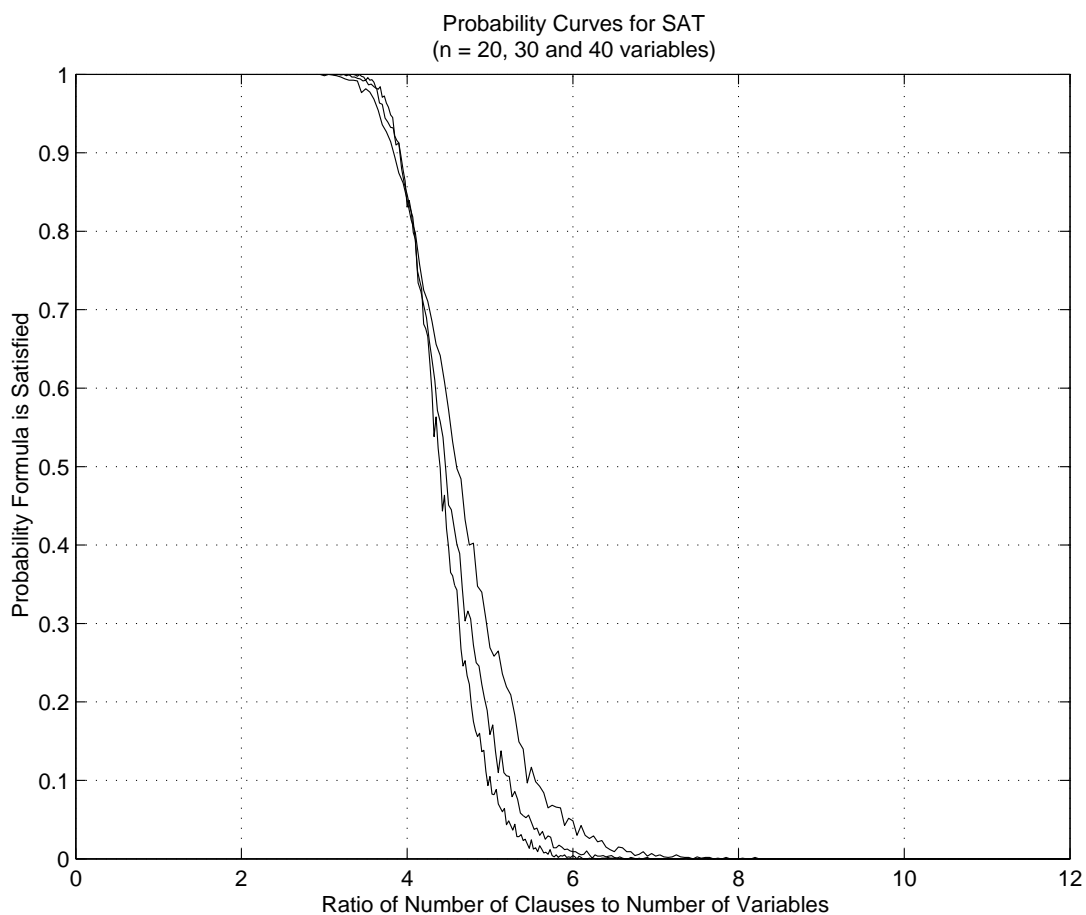


Figure 2.2: Probability curves for 3-SAT for $n = 20, 30, 40$ variables

Selman, Mitchell and Levesque [SML96] for random 3CNF-formulas also revealed that the critical ratio 4.2 appears to be the location at which the average cost of the Davis-Putnam-Logemann-Loveland (DPLL) procedure for Boolean satisfiability peaks. 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.

Figures 2.2 and 2.3 show an example of these phenomena.

Kirkpatrick and Selman [KS94b] noted the similarity of these phenomena to effects in statistical physics and the behavior of spin glasses and, borrowing techniques from that area, used finite-size scaling to arrive at an equation which fit their observations well.

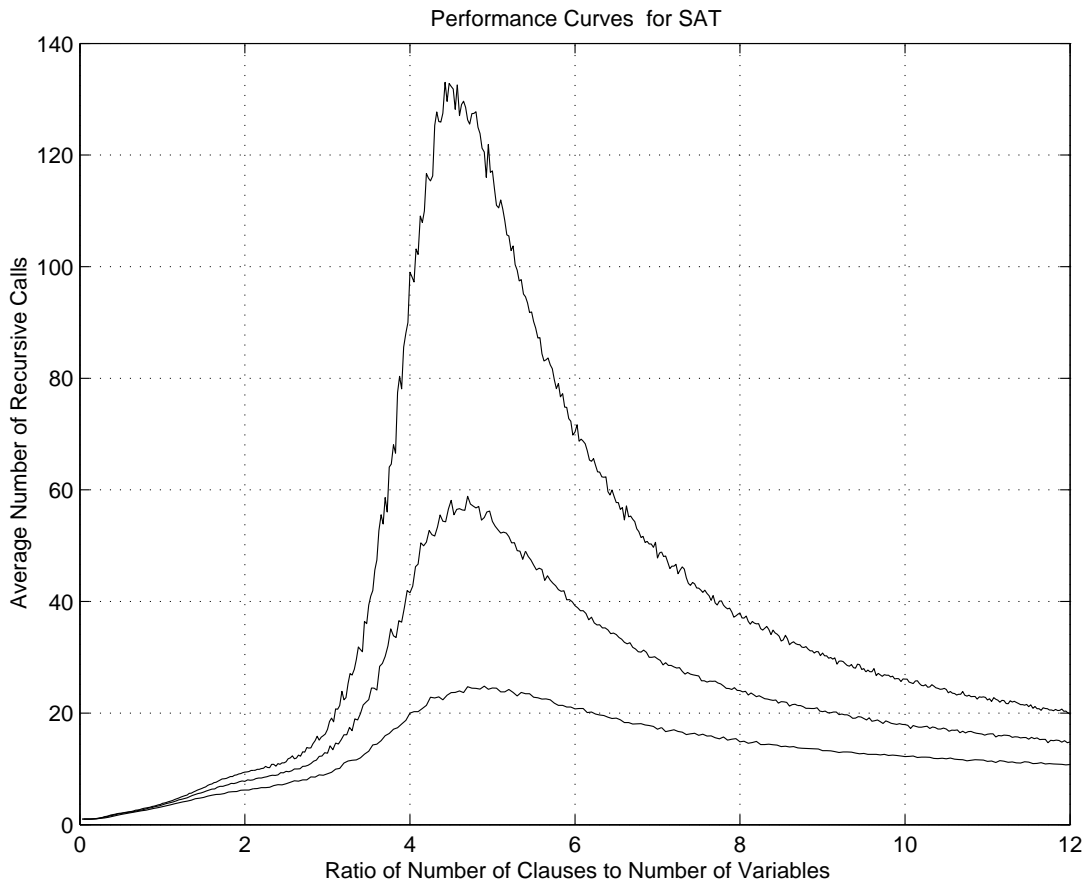


Figure 2.3: Performance curves for 3-SAT for $n = 20, 30, 40$ variables

On the analytical side for $k \geq 3$ progress has been made in establishing tighter and tighter bounds for the value of the critical r_3 , assuming it exists, and in quantifying the sharpness of the threshold in the transition region. The upper bounds are usually derived from a combinatorial/probabilistic approach and the lower bounds from an existence proof of an algorithm that is almost surely guaranteed to find a satisfying assignment. The nature of the threshold has been analyzed with techniques borrowed from random graph theory.

The simplest upper bound for k -SAT can be found with the technique of indicator variables for counters and follows easily from the linearity of expectation and a first moment argument. We will use the following notation:

$V = \{x_1, x_2, x_3, \dots, x_n\}$	a finite set of variables.
$\sigma = (t_1, t_2, t_3, \dots, t_n)$	a truth assignment to the variables in V .
$\Sigma_n = \{\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_{2^n}\}$	the set of all possible truth assignments on n variables.
$\mathcal{C}_n^k = \{d_1, d_2, d_3, \dots, d_{ \mathcal{C}_n^k }\}$	set of all possible k -literal disjuncts formed from V .
φ	a Boolean formula.
$ \varphi $ or $\mathcal{N}_\sigma(\varphi)$	the number of satisfying assignments of φ .
$F_k(n, m)$	the family of m -clause CNF formulas which can be formed by concatenating m randomly selected clauses from \mathcal{C}_n^k without replacement.
$P(A)$	the probability of event A .
$I(A)$	the indicator function for event A .
$E(X)$	the expectation of random variable V .

Note that: the total number of possible truth assignments on n variables is:

$$|\Sigma_n| = 2^n,$$

the total number of distinct clause types is:

$$|\mathcal{C}_n^k| = \binom{n}{k} 2^k$$

and the number of syntactically distinct formulas is:

$$|\mathcal{F}_{n,m}^k| = \left(\binom{n}{k} 2^k \right)^m$$

Since disjunct types may be repeated in a syntactical formula there may be many formulas that are different syntactically but equivalent logically, i.e., contain exactly the same set of disjuncts. Depending on m , the total number of syntactical formulas is arbitrarily large. The maximum possible number of logically distinct formulas depends only on n and k , regardless of how large m is.

$$|\mathcal{P}\{\mathcal{C}_n^k\}| = 2^{|\mathcal{C}_n^k|} = 2^{\binom{n}{k} 2^k}$$

If m is sufficiently large, $m \geq |\mathcal{C}_n^k|$, all of these can be realized.

Now we can state and prove the basic upper bound theorem:

Theorem 2.2.2: (*Markov Upper Bound*) For rational, $r > 0$, integers, $k \geq 2$, $n \geq 1$, and a random formula, φ drawn uniformly from $F_k(n, \lceil nr \rceil)$,

$$r > \frac{1}{k - \lg(2^k - 1)} \text{ implies } \lim_{n \rightarrow \infty} \mathbf{P}(\varphi \text{ is satisfiable}) = 0.$$

Proof: Define $|\varphi|$ as the random variable indicating the number of satisfying assignments of a randomly selected formula from $F_k(n, nr)$, then $\mathbf{P}(\varphi \text{ is satisfiable}) = \mathbf{P}(|\varphi| \geq 1)$. Note that

$$|\varphi| = \sum_{\sigma \in \Sigma_n} I(\sigma \text{ satisfies } \varphi)$$

and from the linearity of expectation,

$$\mathbf{E}(|\varphi|) = \sum_{\sigma \in \Sigma_n} \mathbf{E}(I(\sigma \text{ satisfies } \varphi))$$

Now, the expectation of an indicator function is the probability of the event it indicates. Note that for any σ , out of all possible $2^k \binom{n}{k}$ clause types there are exactly $\binom{n}{k}$ that it does not satisfy and it

satisfies the rest. So, there are $(2^k \binom{n}{k} - \binom{n}{k})^{\lceil nr \rceil}$ formulas in the space that it satisfies out of a total of $(2^k \binom{n}{k})^{\lceil nr \rceil}$. Thus for any σ ,

$$\mathbb{E}(I(\sigma \text{ satisfies } \varphi)) = \mathbb{P}(\sigma \text{ satisfies } \varphi) = \frac{(2^k \binom{n}{k} - \binom{n}{k})^{\lceil nr \rceil}}{(2^k \binom{n}{k})^{\lceil nr \rceil}} = \left(\frac{2^k - 1}{2^k} \right)^{\lceil nr \rceil},$$

and it follows that

$$\mathbb{E}(|\varphi|) = \sum_{\sigma \in \Sigma_n} \left(\frac{2^k - 1}{2^k} \right)^{\lceil nr \rceil} = 2^n \left(\frac{2^k - 1}{2^k} \right)^{\lceil nr \rceil} \leq 2^n \left(\frac{2^k - 1}{2^k} \right)^{rn}$$

Now we have the expectation we can use the first moment method to bound the probability. Markov's inequality gives,

$$\mathbb{P}(|\varphi| \geq 1) \leq \mathbb{E}(|\varphi|) \leq 2^n \left(\frac{2^k - 1}{2^k} \right)^{rn}$$

and it follows that

$$\lim_{n \rightarrow \infty} \mathbb{P}(\varphi \text{ is satisfiable}) \leq \lim_{n \rightarrow \infty} \left(2 \left(\frac{2^k - 1}{2^k} \right)^r \right)^n$$

and a sufficient condition for this limit to go to zero is

$$\begin{aligned} 2 \left(\frac{2^k - 1}{2^k} \right)^r &< 1 \\ 1 + r \lg \left(\frac{2^k - 1}{2^k} \right) &< 0 \\ -1 + r \lg \left(\frac{2^k}{2^k - 1} \right) &> 0 \\ r &> \frac{1}{\lg \left(\frac{2^k}{2^k - 1} \right)} = \frac{1}{k - \lg(2^k - 1)} \end{aligned}$$

■

2.3 Random Structures and Threshold Phenomena

2.3.1 Random Sets and Random Graphs

This is a brief statement of some of the basics. For more extensive treatments see the excellent texts of Bollobás [Bol86] and of Anderson [And89].

Let $X = \{1, 2, 3, \dots, n\}$, an n -set, and consider various $\mathcal{F} \subset \mathcal{P}(X)$, i.e., families of subsets of X .

Let $\mathcal{X}^{(m)} = \{x \in \mathcal{P}(X) : |x| = m\}$. These sets, one for each $0 \leq m \leq n$, the collections of subsets of X which all have the equivalent cardinality, are called *level* sets and for a specific level, say m , the sets in the collection are called m -subsets. Clearly, $\mathcal{X}^{(0)} = \{\emptyset\}$, $\mathcal{X}^{(n)} = \{X\}$ and $|\mathcal{X}^{(m)}| = \binom{n}{m}$.

Let A be a subset of $\mathcal{X}^{(m)}$ for, $m < n$, the set $\nabla A = \{x \in \mathcal{X}^{(m+1)} : A \subset x\}$ is called the *upper shadow* of A and for, $m > 1$, the set $\Delta A = \{x \in \mathcal{X}^{(m-1)} : x \subset A\}$ is called the *lower shadow* of A .

Any *property* that is associated with subsets of $\mathcal{P}(X)$ can be identified with the family of sets, say $\mathcal{Q} \subset \mathcal{P}(X)$, that have the property. And we define the set of level sets of size m that have the property as $\mathcal{Q}^{(m)} = \mathcal{Q} \cap \mathcal{X}^{(m)}$. We will call the property *trivial* if either $\mathcal{Q} = \emptyset$ or $\mathcal{Q} = X$. A property, \mathcal{Q} , is *monotone increasing* if $A \in \mathcal{Q}$ and $A \subseteq B$ implies $B \in \mathcal{Q}$ and is *monotone decreasing* if $A \in \mathcal{Q}$ and $A \supseteq B$ implies $B \in \mathcal{Q}$. It follows then that a monotone increasing property is *non trivial* iff $\emptyset \notin \mathcal{Q}$ and $X \in \mathcal{Q}$ and likewise that a monotone decreasing property is *non trivial* iff $\emptyset \in \mathcal{Q}$ and $X \notin \mathcal{Q}$.

We can define a probability space on $\mathcal{X}^{(m)}$ in which each one of the m -sets is equally probable as follows:

$$\Omega = \mathcal{X}^{(m)}$$

$$\mathcal{A} = \mathcal{P}(\Omega)$$

$$\mathbf{P}(A) = \sum_{\omega \in A} p(\omega) \text{ for all } A \in \mathcal{A}$$

where

$$p(\omega) = \frac{1}{|X^{(m)}|} = \frac{1}{\binom{n}{m}}$$

We will call this model one, and for particular n and m , refer to it as $M_I(n, m)$.

Now let $P_{n,m}(\mathcal{Q})$ be the probability that a random m -set has the property \mathcal{Q} . Clearly

$$P_{n,m}(\mathcal{Q}) = \sum_{\omega \in \mathcal{Q}^{(m)}} \frac{1}{\binom{n}{m}} = \frac{|\mathcal{Q}^{(m)}|}{\binom{n}{m}}$$

Theorem 2.3.1: For all m and \mathcal{Q} such that $0 < m < n$ and $\mathcal{Q} \subset \mathcal{P}(X)$, $P_{n,m}(\mathcal{Q}) \leq P_{n,m+1}(\mathcal{Q})$.

Proof:

It suffices to show that:

$$\frac{|\mathcal{Q}^{(m)}|}{\binom{n}{m}} \leq \frac{|\mathcal{Q}^{(m+1)}|}{\binom{n}{m+1}}$$

It is not immediately obvious that this is so, because with respect to m , $|\mathcal{Q}^{(m)}|$ is monotone decreasing and $\binom{n}{m}$ is not monotone at all.

This difficulty can be avoided by considering sets of permutations of the elements of X . Let $\mathcal{X}^{\pi(m)} = \{s : s = (s_1, s_2, s_3, \dots, s_m), s_i \in X, i \neq j \text{ implies } s_i \neq s_j\}$, the permutations of the elements of X taken m at a time. Clearly, $|\mathcal{X}^{\pi(m)}| = |\mathcal{X}^{(m)}|m!$. Now there are two natural mappings relating sets and permutations of their elements. Let

$$\text{perms}(x) = \{\pi : \pi \text{ is a permutation of the elements of the set } x\} \quad \text{and}$$

$$\text{set_of}(\pi) = \{x : x \text{ is an element of the permutation } \pi\}.$$

Now consider $\mathcal{Q}^{(m)}$, and the sets of permutations of the elements of its members. There are exactly $m!$ permutations for each of the members and there are no permutations among these that are common to any two members of $\mathcal{Q}^{(m)}$. It follows that:

$$|\{\pi : \text{set_of}(\pi) \in \mathcal{Q}^{(m)}\}| = |\mathcal{Q}^{(m)}|m!$$

and of course,

$$|\mathcal{X}^{\pi(m)}| = |\mathcal{X}^{(m)}|m!$$

so

$$\frac{|\{\pi : \text{set_of}(\pi) \in \mathcal{Q}^{(m)}\}|}{|\mathcal{X}^{\pi(m)}|} = \frac{|\mathcal{Q}^{(m)}|m!}{|\mathcal{X}^{(m)}|m!} = \frac{|\mathcal{Q}^{(m)}|}{|\mathcal{X}^{(m)}|}$$

That is, the ratio of the m -permutations that are related to a set property is equivalent to the ratio of m -sets that have the property. Consequently we will have the result we are looking for if we can show

$$\frac{|\{\pi \in \mathcal{X}^{\pi(m)} : \text{set_of}(\pi) \in \mathcal{Q}^{(m)}\}|}{|\mathcal{X}^{\pi(m)}|} \leq \frac{|\{\pi \in \mathcal{X}^{\pi(m+1)} : \text{set_of}(\pi) \in \mathcal{Q}^{(m+1)}\}|}{|\mathcal{X}^{\pi(m+1)}|} \quad (2.1)$$

Let $\pi = (\pi_1, \pi_2, \pi_3, \dots, \pi_m)$ and $\text{pre}(k, \pi) = (\pi_1, \pi_2, \pi_3, \dots, \pi_k)$ where $k \leq m$. We call this the k -prefix of π . Clearly, since \mathcal{Q} is monotonic increasing, for all $\pi \in \mathcal{X}^{\pi(m+1)}$,

$$\text{set_of}(\text{pre}(m, \pi)) \in \mathcal{Q}^{(m)} \text{ implies } \text{set_of}(\pi) \in \mathcal{Q}^{(m+1)}.$$

so

$$|\{\pi \in \mathcal{X}^{\pi(m)} : \text{set_of}(\pi) \in \mathcal{Q}^{(m)}\}|(n-m) \leq |\{\pi \in \mathcal{X}^{\pi(m+1)} : \text{set_of}(\pi) \in \mathcal{Q}^{(m+1)}\}|$$

and of course

$$|\mathcal{X}^{\pi(m)}|(n-m) = |\mathcal{X}^{\pi(m+1)}|$$

and equation (2.1) immediately follows. ■

A tighter inequality, known as the local LYM inequality [Lub66, Yam54, Mes63] can be shown in which the probability of Q on the $m + 1$ level can be replaced by the probability of the $m + 1$ level sets that contain sets in $Q^{(m)}$. This is possibly smaller since with a monotone increasing property there can be $Q^{(m+1)}$ sets that do not contain any $Q^{(m)}$ sets.

Theorem 2.3.2: *If A is a subset of $\mathcal{X}^{(m)}$ where $m \leq n - 1$, then $|A|/\binom{n}{m} \leq |\nabla A|/\binom{n}{m+1}$.*

Proof: Let $(x, y) \in \mathcal{X}^{(m)} \times \mathcal{X}^{(m+1)}$ and consider the two relations:

$$R_1 = \{(x, y) : x \in A \text{ and } y \supseteq x\}$$

$$R_2 = \{(x, y) : y \in \nabla A \text{ and } y \supseteq x\}$$

Since $(x \in A \text{ and } y \supseteq x)$ implies $(y \in \nabla A \text{ and } y \supseteq x)$, it must be the case that $R_1 \subseteq R_2$ and it follows that

$$|R_1| \leq |R_2|$$

Now for each $x \in A$ there are exactly $\binom{n-m}{1}$, $y \in \mathcal{X}^{(m+1)}$, that contain it, and on the other hand, for each $y \in \nabla A$ there are exactly $\binom{m+1}{1}$, $x \in \mathcal{X}^{(m)}$, that it contains. So we have:

$$|R_1| = |A|(n - m)$$

$$|R_2| = |\nabla A|(m + 1)$$

It follows then that:

$$|A|(n - m) \leq |\nabla A|(m + 1)$$

$$|A| \leq |\nabla A| \frac{m + 1}{n - m} = |\nabla A| \frac{\binom{n}{m}}{\binom{n}{m+1}}$$

$$\frac{|A|}{\binom{n}{m}} \leq \frac{|\nabla A|}{\binom{n}{m+1}}$$

■

In the following we will find it convenient to extend the definition of the binomial coefficient to handle reals in the usual way. Let

$$\binom{x}{n} = \frac{x(x-1)(x-2)\dots(x-n+1)}{n!}$$

where x is a positive real number and n is a non-negative integer.

Theorem 2.3.3 *Let $A \subset X^{(k)}$ be a non-empty collection of k -sets, with $k \geq 1$, let $m > 0$ be an integer and define real x by $x \geq m$ and $|A| = \binom{x}{m}$, then*

$$|\Delta A| \geq \binom{x}{m-1}$$

This result due to Lovász [Lov79] is closely related to the Kruskal-Katona theorem and gives the tightest bound on the cardinality of the shadow of a k -set with respect to the cardinality of the set itself.

Theorem 2.3.4: (Bollobás) *If Q is a monotone decreasing property and $0 \leq m_1 < m_2 \leq n$ where $n = |X|$, then*

$$P_{m_1}(Q)^{m_2} \geq P_{m_2}(Q)^{m_1}$$

Proof: This closely follows [Bol86] with a few more details. First, consider the extreme cases for $P_{m_2}(Q)$. If $Q_{m_2} = X^{(m_2)}$ then $P_{m_2}(Q) = 1$ and clearly since Q is monotone decreasing $P_{m_1}(Q)$ must also equal 1, so equation 2.3.4 holds. On the other hand, if $Q_{m_2} = \emptyset$ then $P_{m_2}(Q) = 0$ and 2.3.4 must hold regardless of the value of $P_{m_1}(Q)$.

The remaining case is $\emptyset \subsetneq Q_{m_2} \subsetneq X^{(m_2)}$. Here we will show that

$$P_{m_1}(Q)^{m_2} \geq \left\{ \frac{\binom{x_1}{m_1}}{\binom{n}{m_1}} \right\}^{m_2} \geq \left\{ \frac{\binom{x_1}{m_2}}{\binom{n}{m_2}} \right\}^{m_1} = P_{m_2}(Q)^{m_1} \quad (2.2)$$

where x_1 is defined by $\binom{x_1}{m_2} = |Q_{m_2}|$.

To prove the first inequality in 2.2, we note that from 2.3.3 we have

$$|\Delta Q_{m_2}| \geq \binom{x_1}{m_2 - 1}$$

and defining x_2 by $\binom{x_2}{m_2-1} = |Q_{m_2-1}|$. It follows that x_2 must be greater than or equal to x_1 . Then applying 2.3.3 again we get

$$|\Delta(\Delta Q_{m_2})| \geq \binom{x_2}{m_2-2}$$

and we can continue this way, and in general, find that

$$|\Delta^i Q_{m_2}| \geq \binom{x_i}{m_2-i} \text{ and } x_i \geq x_{i-1} \geq \dots \geq x_1 \quad (2.3)$$

where we define $\Delta^1(\cdot) = \Delta(\cdot)$ and for $i > 1$, $\Delta^i(\cdot) = \Delta(\Delta^{i-1}(\cdot))$. Now setting $i = m_2 - m_1$ in 2.3 and noting that from the monotonicity of Q , all shadows of sets in Q must also be in Q , we have

$$P_{m_1}(Q) \geq P_{m_1}(\Delta^{m_2-m_1}(Q_{m_2})) = \frac{|\Delta^{m_2-m_1}(Q_{m_2})|}{\binom{n}{m_1}} \geq \frac{\binom{x_{m_2-m_1}}{m_1}}{\binom{n}{m_1}} \geq \frac{\binom{x_1}{m_1}}{\binom{n}{m_1}}$$

from which it immediately follows

$$P_{m_1}(Q)^{m_2} \geq \left\{ \frac{\binom{x_1}{m_1}}{\binom{n}{m_1}} \right\}^{m_2}$$

To prove the second inequality in 2.2 we use the fact that: If $0 \leq k \leq x \leq n$, then $(x-k)/(n-k)$ is monotone decreasing for increasing k .

$$\begin{aligned} \left(\frac{x_1}{n}\right)^{m_2-m_1} \dots \left(\frac{x_1-m_1+1}{n-m_1+1}\right)^{m_2-m_1} &\geq \left(\frac{x_1-m_1+1}{n-m_1+1}\right)^{(m_2-m_1)m_1} \\ &\geq \left(\frac{x_1-m_1}{n-m_1}\right)^{m_1(m_2-m_1)} \\ &\geq \left(\frac{x_1-m_1}{n-m_1}\right)^{m_1} \dots \left(\frac{x_1-m_2+1}{n-m_2+1}\right)^{m_1} \\ \left(\frac{x_1}{n}\right)^{m_2} \dots \left(\frac{x_1-m_1+1}{n-m_1+1}\right)^{m_2} &\geq \left(\frac{x_1}{n}\right)^{m_1} \dots \left(\frac{x_1-m_2+1}{n-m_2+1}\right)^{m_1} \\ \left\{ \frac{\binom{x_1}{m_1}}{\binom{n}{m_1}} \right\}^{m_2} &\geq \left\{ \frac{\binom{x_1}{m_2}}{\binom{n}{m_2}} \right\}^{m_1} \end{aligned}$$

Lastly, the remaining equality statement in 2.2 is immediate from definition.

■

2.3.2 Threshold Functions

A sequence $m^*(n)$ is a *threshold function* for Q if for any function $m(n)$ such that

$$\frac{m(n)}{m^*(n)} \rightarrow \infty \text{ implies } \lim_{n \rightarrow \infty} P_{m(n)}(Q) = 0$$

and such that

$$\frac{m(n)}{m^*(n)} \rightarrow 0 \text{ implies } \lim_{n \rightarrow \infty} P_{m(n)}(Q) = 1$$

Theorem 2.3.5: (Bollobás and Thompson) *Every monotone decreasing set property has a threshold function.*

Proof: The proof is constructive. Let Q be a monotone decreasing property defined on the family of sets $X^{(n)}$. Then the following function is a threshold function for Q .

$$m^*(n) = \begin{cases} \max m > 0 : P(Q^{(m)}) \geq 1/2, & \text{if it exists;} \\ 1, & \text{otherwise.} \end{cases} \quad (2.4)$$

Let $m(n)$ be any positive integer valued function of n . Then from theorem 2.3.4 whenever $m(n) > m^*(n) + 1$ we have:

$$P_{m(n)}(Q) \leq P_{m^*(n)+1}(Q)^{\frac{m(n)}{m^*(n)+1}} \leq \left(\frac{1}{2}\right)^{\frac{m(n)}{m^*(n)+1}} < \left(\frac{1}{2}\right)^{\frac{m(n)}{m^*(n)+1}}$$

and

$$\frac{m(n)}{m^*(n)} \rightarrow \infty \text{ and } m^*(n) \geq 1 \text{ implies } \frac{m(n)}{m^*(n) + 1} \rightarrow \infty$$

therefore,

$$\frac{m(n)}{m^*(n)} \rightarrow \infty \text{ implies } \lim_{n \rightarrow \infty} P_{m(n)}(Q) = 0$$

The first condition for a threshold function. On the other hand, whenever $m(n) < m^*(n)$ we have:

$$P_{m(n)}(Q) \geq P_{m^*(n)}(Q)^{\frac{m(n)}{m^*(n)}} \geq \left(\frac{1}{2}\right)^{\frac{m(n)}{m^*(n)}} \geq \left(\frac{1}{2}\right)^{\frac{m(n)}{m^*(n)}}$$

therefore,

$$\frac{m(n)}{m^*(n)} \rightarrow 0 \text{ implies } \lim_{n \rightarrow \infty} P_{m(n)}(Q) = 1$$

The second condition for a threshold function. ■

An intuitive description of what the threshold function defined in the proof does may be glimpsed from the following. For a fixed n , consider the graph of the one-variable function $P_n(m) = P(n, m)$; the vertical axis is the positive reals and the horizontal axis is the positive integers, and $P_n(m)$ is monotone decreasing function of m with a supremum of 1 and an infimum of 0. Now the threshold function $m^*(n)$, for fixed n , is just a constant and we can rescale the horizontal axis by dividing by it, say $r = x/m_n^*$, giving us a new axis, r . This is a kind normalization that forces the 1/2 value of the probability curve to occur at $r = 1$ for the transformed curve. We can do this for each value of n and plot all the $P_n(r)$ curves on the same graph. Then all of the curves will not only be monotonic decreasing but they will all cross at $r = 1$. (Note that m_n for different values of n are not necessarily all the same.) Now we can state the threshold property in terms of an arbitrary sequence of points on the r -axis rather than in terms of arbitrary $m(n)$ functions. Take any sequence of points on the r -axis, say their values are $r(n)$, the threshold property says that if the sequence tends to zero, the sequence of values $P_n(r(n))$, i.e., the height of the n -th probability curve above the point $r(n)$, tends to one, and likewise, if the sequence of points tends to infinity, the probability values tend to zero.

It is clear that threshold functions are not unique in the sense that if $m^*(n)$ is a threshold function for a property Q then any function that is $\Theta(m^*(n))$ is also a threshold function for Q . For example if the threshold function that makes the curves for some Q cross at 1/2 upon rescaling happened to be $m^*(n) = n + \sin(\frac{2\pi n}{100})$ then certainly $m^*(n) = n$ would be a simpler and perfectly good threshold function and the crossover points for the curves would oscillate if that function were used for rescaling.

Another common probability model used in studies of families of subsets of an n -set, X , is to consider the random sets which are constructed by a series of inclusion decisions in which each member of X is included with a fixed probability, say p . In this case the probability space is defined as follows:

$$\Omega = X$$

$$\mathcal{A} = \mathcal{P}(\Omega)$$

$$\mathbf{P}(A) = p^{|A|}(1-p)^{n-|A|} \quad \text{for all } A \in \mathcal{A}$$

We will call this model two, and for particular n and p refer to it as $M_{II}(n, p)$ and define $P_{n,p}$ as the probability that random set from this model has property Q .

A special case of this model is when $p = 1/2$. For this, since $p = 1 - p$ the probability for all of the subsets is the same, namely, p^n , i.e., $1/2^n$. Formally:

$$\Omega = X$$

$$\mathcal{A} = \mathcal{P}(\Omega)$$

$$\mathbf{P}(A) = \frac{1}{2^n} \quad \text{for all } A \in \mathcal{A}$$

This is a common model for random graphs where all the possible graphs on N vertices are assigned equal probability. For this, the base set X is the set of all possible edges and $n = \binom{N}{2}$. Erdős introduced this model for graphs and studied it extensively [Erd47].

For the general case with $0 < p < 1$ it is easy to show that the expected size of the random subsets of $M_{II}(n, p)$ is np . Furthermore, with respect to monotone set properties the asymptotic behavior of $M_I(n, np)$ and $M_{II}(n, p)$ are the same in the senses of the following two propositions and theorem [JLR00]:

Proposition 2.3.6: Let X be an n -set, Q be a property on $\mathcal{P}(X)$, $M_I(n, m)$ and $M_{II}(n, p)$ be the model I and model II probability spaces associated with X and let $X(n, m)$ and $X(n, p)$ be the random set variables that correspond to drawing a set at random from these spaces, let a sequence $p = p(n) \in [0, 1]$ and a fixed real $a \in [0, 1]$, then, for every sequence $m = m(n)$ such that $m = np + O(\sqrt{np(1-p)})$,

$$\lim_{n \rightarrow \infty} \mathbb{P}(X(n, m) \in Q) = a$$

implies

$$\lim_{n \rightarrow \infty} \mathbb{P}(X(n, p) \in Q) = a$$

Proposition 2.3.7: Let X be an n -set, Q be a monotone property on $\mathcal{P}X$, $M_I(n, m)$ and $M_{II}(n, p)$ be the model I and model II probability spaces associated with X and let $X(n, m)$ and $X(n, p)$ be the random set variables that correspond to drawing a set at random from these spaces, let a sequence $m = m(n)$ such that $0 \leq m \leq n$ and a fixed real $a \in [0, 1]$, then, for every sequence $p = p(n)$ such that $p = m/n + O(\sqrt{m(n-m)/n^3})$,

$$\lim_{n \rightarrow \infty} \mathbb{P}(X(n, p) \in Q) = a$$

implies

$$\lim_{n \rightarrow \infty} \mathbb{P}(X(n, m) \in Q) = a$$

Theorem 2.3.8: Let X be an n -set, Q be a monotone increasing property on $\mathcal{P}X$, $M_I(n, m)$ and $M_{II}(n, p)$ be the model I and model II probability spaces associated with X and let $X(n, m)$ and $X(n, p)$ be the random set variables that correspond to drawing a set at random from these spaces, then

$$\text{If } \lim_{n \rightarrow \infty} \mathbb{P}(X(n, p) \in Q) = 1 \text{ then } \lim_{n \rightarrow \infty} \mathbb{P}(X(n, pn) \in Q) = 1$$

$$\text{If } \lim_{n \rightarrow \infty} \mathbb{P}(X(n, p) \in Q) = 0 \text{ then } \lim_{n \rightarrow \infty} \mathbb{P}(X(n, pn) \in Q) = 0$$

If $\lim_{n \rightarrow \infty} \mathbb{P}(X(n, pn) \in Q) = 1$ then $\lim_{n \rightarrow \infty} \mathbb{P}(X(n, p(1 + \delta)) \in Q) = 1$

If $\lim_{n \rightarrow \infty} \mathbb{P}(X(n, pn) \in Q) = 0$ then $\lim_{n \rightarrow \infty} \mathbb{P}(X(n, p(1 - \delta)) \in Q) = 0$

The above two propositions and theorem can be proven using the theory of total probability and Chebychev's inequality [JLR00]. These results make it possible use either model I or model II, which ever happens to be most convenient, when investigating the asymptotic behavior of set properties.

2.3.3 Coarse and Sharp Thresholds

Given the model $M_{II}(n, p)$, random set variable $X(n, p)$, a sequence $p = p(n) \in [0, 1]$. and a non-trivial monotone decreasing property Q , we define $\mu(Q, n, p) = \mathbb{P}(X(n, p) \in Q)$ and $\mu_x(p) = p$ such that $\mu(Q, n, p) = x$. And we say Q has a *model II sharp threshold* if for any $0 < \epsilon < 1/2$,

$$\lim_{n \rightarrow \infty} \frac{\mu_{1-\epsilon}(p) - \mu_{\epsilon}(p)}{\mu_{\frac{1}{2}}(p)} = 0$$

If the limit is bounded away from zero, we say Q has a *coarse threshold*. Theorem 2.3.5 shows this limit is bounded above.

This model II sharp threshold definition is the usual definition for sharp threshold.

It is also possible to define a sharp threshold in the context of model I: Given the model $M_I(n, m)$, random set variable $X(n, m)$, a sequence $m = m(n)$ with $0 \leq m \leq n$, and a non-trivial monotone decreasing property Q , we define $\mu(Q, n, m) = \mathbb{P}(X(n, m) \in Q)$ and $\mu_x(m) = m$ such that $\mu(Q, n, m) = x$. And we say Q has a *model I sharp threshold* if for any $0 < \epsilon < 1/2$,

$$\lim_{n \rightarrow \infty} \frac{\mu_{1-\epsilon}(m) - \mu_{\epsilon}(m)}{\mu_{\frac{1}{2}}(m)} = 0$$

If the limit is bounded away from zero, we say Q has a *coarse threshold*.

Proposition 2.3.9: *Given $M_I(n, m)$ and $M_{II}(n, p)$ as defined in above in the definitions of sharp thresholds for models I and II, if $m = np$, then the definitions are equivalent, i.e.,*

Proof: It follows immediately, when the above propositions hold, that $n(\mu_x(p)) = \mu_x(np)$ and then it is clear that

$$\lim_{n \rightarrow \infty} \frac{\mu_{1-\epsilon}(m) - \mu_\epsilon(m)}{\mu_{\frac{1}{2}}(m)} = \lim_{n \rightarrow \infty} \frac{\mu_{1-\epsilon}(p) - \mu_\epsilon(p)}{\mu_{\frac{1}{2}}(p)}.$$

Models I and II are extensively used in random graph investigations. In that context X is taken to be the set of all possible edges on a labeled vertex set. For example, without loss of generality let $V = \{1, 2, 3, \dots, n\}$, then $X = \{\{i, j\} | i, j \in V \text{ and } i \neq j\}$ and define $G(n, m)$ to be $M_I(\binom{n}{2}, m)$ and $G(n, p)$ to be $M_{II}(\binom{n}{2}, p)$. Customarily, in addition to representing a random graph model, $G(n, m)$ and $G(n, p)$ are used to represent the random graph variables associated with the random graph models.

When using the random set/graph concepts and methods to deal with random Boolean formulas the usual approach is to let X be the set of all possible clauses, i.e., $X = \mathcal{C}_n^k$. When this is done, it should be noted that model $M_I(n, m)$ is not exactly the same as $F_k(n, m)$. In the former, there are no repetitions of clauses, while in the latter there may be. If you think of generating the random formulas by uniformly drawing clauses from \mathcal{C}_n^k , then in one case, it is sampling without replacement and in the other, sampling with replacement. This discrepancy does not appear to be dealt with in the literature. In an unpublished paper, Dalmau has shown that, with respect to the existence of a sharp threshold, the model with replacement is equivalent to the one without replacement. Moreover, for $F_k(n, m)$ with $k > 2$ the following theorem shows that, with respect to all asymptotic properties, randomly selecting the clauses, with or without replacement is equivalent.

Theorem 2.3.10 *For all $k > 2$, the probability of a fixed ratio (number of clauses/number of variables) random k -CNF formula having no repeating clauses tends to one as the number of variables tends to infinity.*

Proof: Let n be the number of variables, then the number of different possible clause types is:

$$N = 2^k \binom{n}{k}$$

Let r be the fixed ratio of clauses-to-variables, then a random formula is generated by selecting rn clauses uniformly with replacement from the N possible clause types. The probability that the i -th clause selected is not a repetition of a previous one is:

$$\mathcal{P}\{i\text{-th selection is not the same as any previous selection}\} = \frac{N - i + 1}{N}$$

and the probability of not having a repetition in the complete formula can be computed as the product of the probabilities that each selection does not repeat a previous one.

$$\begin{aligned} \mathcal{P}\{\text{all clauses different}\} &= \left(\frac{N}{N}\right) \left(\frac{N-1}{N}\right) \left(\frac{N-2}{N}\right) \cdots \left(\frac{N-rn+1}{N}\right) \\ &> \left(1 - \frac{rn-1}{N}\right)^{rn-1} \\ &= \left(e^{-\frac{rn-1}{N}} + o(1)\right)^{rn-1} \\ &> e^{-\frac{(rn-1)^2}{N}} \end{aligned}$$

and since, $(rn - 1)^2$ is $\Theta(n^2)$ and N is $\Theta(n^k)$, it immediately follows that if $k > 2$

$$\lim_{n \rightarrow \infty} \mathcal{P}\{\text{all clauses different}\} = 1$$

■

Friedgut [Fri99] proved that for all $k \geq 2$, k -SAT has a sharp threshold. Specifically, expressed in our notation, he proved:

Theorem 2.3.11: (Friedgut) *For every fixed $k \geq 2$, there exists a function $r(n)$ such that for every $\epsilon > 0$, where $\varphi(m)$ is a random formula drawn uniformly from $F_k(n, m)$,*

$$\lim_{n \rightarrow \infty} \mathbb{P}(\varphi(\lceil n(r - \epsilon) \rceil) \text{ is satisfiable}) = 1$$

$$\lim_{n \rightarrow \infty} \mathbb{P}(\varphi(\lceil n(r + \epsilon) \rceil) \text{ is satisfiable}) = 0$$

Note that this does not state that there is a phase transition, since the function, $r(n)$, may or may not have a limit. This is still an open question.

2.4 General Experimental Methodology for this Research

All of the experiments in this investigation are “computer experiments”, i.e., characteristics of specified probability spaces are estimated as measures of computer-simulated random samples. Sample size is a major parameter in this type of work. We used a sample size of 1200 in all of our experiments. With respect to structural phase transitions, the parameter that we are interested in is the fraction of formulas in some finite space that have some specified property, e.g., satisfiability. This fits the model of Bernoulli trials in a binomial distribution where we want to estimate the probability of success. The sample size of 1200 yields a 95% confidence that the average number of successful trials in our experiments is within plus or minus 0.03 of the true fraction.

With respect to algorithmic cost the picture is somewhat different. Two questions that both admit of several reasonable answers must be addressed. First, how should cost be measured for individual samples? Second, what statistical measure is appropriate for describing the sample as a whole, i.e., mean, median etc.? In cases where it is easy to determine, we use a machine-performance independent measure of algorithm cost. For the DPLL, this is the number of recursive calls required to solve the problem. In cases where it not convenient to get the number of recursive calls, e.g., working with a packaged algorithm, we use actual clock time. This type of measurement makes it difficult to compare results from different environments quantitatively, but qualitatively it is useful as it still shows the dynamics of the problem quite well.

Mean, median and mode are some ways of summarizing the information in the data. None of them tell the whole story and each has distinct strengths and weaknesses as summarizers. In the case of summarizing the performance of an algorithm over a random sample of inputs the mean is particularly useful because of its intrinsic relationship to expectation. If you want to know how long it is going to take to process 1000 random samples, the mean gives you a simple and principled way to do it. Expectation plays a prominent role in probability theory, and there are a wealth of tools for dealing

with it. Next to bounds, the most typical parameter analyzed in performance is expectation. Therefore, since we are also able to work with large enough sample sizes to obtain reasonable estimates for the mean, we use it. Some researchers have found it more convenient to report the median. If sample sizes are small, it may be the case that the sample average may exhibit considerable variance. Also, it is possible to find the sample median for algorithmic cost without running every instance to completion by judiciously picking some value to abort the computation. If the abort value is above the sample's actual median, this process will not affect its computation. However, this method leaves open the question of whether an aborted instance was satisfying or not.

2.5 Decision Versions of Optimization and Counting Problems

Counting problems and many optimization problems are positive integer function problems. That is, they can be cast as: Input: $i \in S$, some instance space. Output: the value $f(i)$ of a function $f : S \rightarrow I$ where $I = \{0, 1, 2, 3, \dots\}$. In general, these are not considered to be capable of directly exhibiting phase transition phenomena since their outputs usually admit of myriad different values. Nevertheless, it is possible to study these problems from the point of view of phase transitions by asking yes or no questions about the value of their outputs. If such a decision problem captures the essence of the function problem, we can think of it as being the decision problem corresponding to the function problem and investigate it with respect to phase transitions.

Consider the decision problem: Input: $i \in S$, some instance space, and $t \in I$ some threshold. Output: the answer to the question: Is $f(i) \geq t$? This decision problem captures the essence of the function problem in the sense that we can easily find the answer to one of the problems if we have an implementation of the other. In one direction, this is trivial and in the other, straightforward. If we know the answer to the function problem, the answer to the decision problem is immediate. If we have an implementation of the decision problem, we can find the value of the function problem with binary

search. The only subtlety is that we have to determine the appropriate range before using simple binary search. The range may be obvious from the description of the problem, but in any case can be found if necessary by an expanding binary search for the smallest power of 2 that is an upper bound.

Part III

Threshold-Counting Satisfiability

Problems

Chapter 3

Threshold-Counting Satisfiability

Problems

3.1 Introduction

Threshold-counting satisfiability problems are a particular kind of generalization of SAT. Given a Boolean formula as an input they ask the question: is the number of satisfying assignments for the formula greater than some threshold amount? The complexity of this type of decision problem depends very much on how the threshold amount is specified. We will primarily consider non-uniform versions of threshold problems. Non-uniform means that the type of threshold and any parameters involved in its specification will be considered to be part of the problem's definition rather than arguments in the problem's input. This allows working with problems that have the same input as SAT, namely just a formula, and when a parameter is involved, allows studying families of such problems. If the threshold is simply the constant, 1, we have just the common Boolean satisfiability problem. On the other hand if we allow the threshold to be specified as some function of n , the number of

variables in the space, we generate a rich collection of threshold problems. We consider here the problems with counting thresholds specified as a constant, a fixed fraction of total possible assignments, and as an exponential portion of total possible assignments. With the exponential type we discover an interesting family of PP-complete satisfiability problems that exhibit phase transitions. The opportunity to study phase transitions in problems that belong to complexity classes beyond NP-complete is important because, although Boolean satisfiability is both a fundamental problem for computer science and is the prototypical NP-complete problem, there are important and practical decision problems that are more complex. Many reasoning and planning problems in artificial intelligence turn out to be complete for complexity classes beyond NP and 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 [Byl94]. A probabilistic analysis of STRIPS planning and an experimental comparison of different algorithms for this problem have been carried out by Bylander [Byl96]. 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 [CGS97, GW99]. 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 problems (or of the complements of all decision problems) computable by nondeterministic polynomial-time Turing machines using oracles from lower levels. See Papadimitriou's text [Pap94] 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 randomized quantifiers applied to a Boolean formula. Experimental results on phase transitions for SSAT have been reported in [Lit99] and [LMP01].

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 [Val79a, Val79b], 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 #P-complete problems arise naturally in logic, algebra, and graph theory [Val79a, Val79b]. Moreover, #P-complete problems are encountered in artificial intelligence; these include the problem of computing Dempster’s rule for combining evidence [Orp90] and the problem of computing probabilities in Bayesian belief networks [Rot96]. 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 by Birnbaum and Lozinskii [BL99]. 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 developed by Bayardo and Pehoushek [BP00]; 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 k SAT, $k \geq 3$, 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 non-deterministic 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 [Sim75] and Gill [Gil77], 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 [LGM98].

It is known that the class PP contains both NP and coNP, and is contained in PSPACE [Pap94]. Moreover, as pointed out by Angluin [Ang80], there is a tight connection between #P and PP. Specifically, $P^{\#P} = P^{PP}$, which means that the class of decision problems computable in polynomial time using #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 #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. Toda [Tod89] showed that $PH \subseteq P^{PP} = P^{\#P}$. 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.

Littman [Lit99], initially carried out experiments to study the median running time of an ex-

tension 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 by Littman et al. [LMP01], 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 Littman et al. [Lit99][LMP01] 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.

3.2 Varieties of Threshold Counting Problems

We are looking for problems more complex than k -SAT that exhibit comparable phase transitions with the goal of understanding more about the formation of the phase transition and its relation to problem difficulty. In this section we explore generalizations of k -SAT from the point of view of considering it to be a special case of counting-threshold problems. The general problem is: given a k -CNF, does it have more than some threshold number of satisfying assignments? Here k -SAT is just the special case of the threshold being specified as 1. Initially we will use a simplified model to get an idea of the behavior of threshold problems and which ones will be suitable for our purpose. The simplified model is based on an approach borrowed from statistical physics, called the Mean Free Field assumption. In the case of k -CNFs it amounts to assuming that there is no interaction between the clauses or, in other words, knowing the truth value of one clause does not give any information about the truth value of another clause.

3.3 Mean Free Field Model

Any particular formula can be thought of as partitioning the set of all possible assignments into two distinct subsets, namely, those assignments that are true under the formula and those assignments that are false under the formula. So an experiment that involves randomly selecting assignments and determining if they satisfy a particular formula consists of Bernoulli trials where the probability of success equals the fraction of all possible assignments that are satisfying. Likewise if we have a way of calculating the probability of success for a particular formula we could use it to find the total number of satisfying assignments for the formula:

$$\begin{aligned} \text{Num of satisfying assignments of } F &= \text{P}(\text{Random assignment satisfies } F)2^n \\ &= \text{P}(\text{Random assignment satisfies } \bigwedge_{i=1}^m c_i)2^n \end{aligned}$$

where c_i are the clauses that make up F . Now if the probability of satisfying any particular clause is independent of the other clauses then:

$$\begin{aligned} \text{P}(\text{Random assignment satisfies } \bigwedge_{i=1}^m c_i) &= \prod_{i=1}^m \text{P}(\text{Random assignment satisfies } c_i) \\ &= \prod_{i=1}^m \left(\frac{2^k - 1}{2^k} \right) \\ &= \left(\frac{2^k - 1}{2^k} \right)^m \end{aligned}$$

The Mean Free Field (MFF) model assumes this independence among the clauses. We of course know that the assumption is not true; however, it is known that this assumption produces useful approximations in the study of large ensembles of interacting systems in physics. We make the assumption and see later how the model relates to reality. Making the independence assumption then we have that:

$$\begin{aligned} \text{Num of satisfying assignments of } F &= \text{P}(\text{Random assignment satisfies } F)2^n \\ &= \text{P}(\text{Random assignment satisfies } \bigwedge_{i=1}^m c_i)2^n \\ &= \left(\frac{2^k - 1}{2^k} \right)^m 2^n \end{aligned}$$

We will see later that this number happens to be the expected value in the real model where independence does not hold. In the MFF model it is the actual number of satisfying assignments for an arbitrary formula. So, whether or not the number of satisfying assignments for a formula of m clauses over n variables is greater than or equal to $t(n)$ depends on whether or not

$$\left(\frac{2^k - 1}{2^k}\right)^m 2^n \geq t(n)$$

From this we find the critical value r^* as follows:

$$\begin{aligned} \left(\frac{2^k - 1}{2^k}\right)^m 2^n &= t(n) \\ \left(\frac{2^k - 1}{2^k}\right)^{r^* n} 2^n &= t(n) \\ \left(\frac{2^k - 1}{2^k}\right)^{r^*} 2 &= t(n)^{\frac{1}{n}} \\ \left(\frac{2^k - 1}{2^k}\right)^{r^*} &= \frac{t(n)^{\frac{1}{n}}}{2} \\ \left(\frac{2^k}{2^k - 1}\right)^{r^*} &= \frac{2}{t(n)^{\frac{1}{n}}} \\ r^* \lg\left(\frac{2^k}{2^k - 1}\right) &= 1 - \frac{\lg(t(n))}{n} \\ r^* &= \frac{1 - \frac{\lg(t(n))}{n}}{k - \lg(2^k - 1)} \end{aligned}$$

So in the MFF model the number of satisfying assignments for a formula is greater than or equal to $t(n)$, if its $r \leq r^*$ and less than $t(n)$, if its $r > r^*$. So if r^* tends to a fixed limit then there will be a phase transition at that point.

The determining factor is what happens to $\lim_{n \rightarrow \infty} \frac{\lg(t(n))}{n}$. Let's look at counting threshold functions of a constant C , a fraction p/q of total possible assignments and an exponential $2^{\alpha n}$.

$$\begin{aligned}
\lim_{n \rightarrow \infty} \frac{\lg C}{n} &= 0 & \text{and } r^* &= \frac{1}{k - \lg(2^k - 1)} \\
\lim_{n \rightarrow \infty} \frac{\lg(\frac{p}{q} 2^n)}{n} &= \lim_{n \rightarrow \infty} \frac{C \lg \frac{p}{q} + n}{n} = 1 & \text{and } r^* &= \frac{1 - 1}{k - \lg(2^k - 1)} = 0 \\
\lim_{n \rightarrow \infty} \frac{\lg(2^{\alpha n})}{n} &= \alpha & \text{and } r^* &= \frac{1 - \alpha}{k - \lg(2^k - 1)}
\end{aligned}$$

We see that for positive counting-threshold functions in this model the critical r^* lies between zero and ≈ 5.19 , the SAT Markov upper bound. If the counting-threshold function grows too rapidly r^* degenerates to zero and if it doesn't grow at all then it goes to the SAT bound. In the zero case, it does not make sense to talk about what happens for $r < r^*$, so from the point of view of phase transitions this is not interesting. From the above, we see this happens for a threshold specified as a fraction of total possible assignments. Actually, in the fixed fraction case, n is not a threshold function at all, so $r = m/n$ is not an appropriate choice for a critical parameter. The $2^{\alpha n}$ function, on the other hand, models a parameterized family of problems with phase transitions whose critical point locations are a monotonic function of the parameter.

3.4 Constant Threshold

Definition 3.4.1: The decision problem #THRESHOLDSAT($\geq c$) asks the following question. Given a k -CNF formula φ as input, does φ have a number of satisfying assignments that is greater than or equal to c , some positive integer constant? ■

The Mean Free Field model suggests that the constant threshold decision problem behaves like SAT. The following shows that this suggestion is true for the actual model. Of course #THRESHOLDSAT(≥ 1) is SAT and although #THRESHOLDSAT($\geq j$) is not the same #THRESHOLDSAT($\geq k$) where $k \neq j$ for all formulas, when we consider the standard random model the asymptotic behavior for these problems is the same. The following theorem highlights the basic property from

which this can be shown.

Theorem 3.4.2: *Given an integer $l \geq 1$, a family of sample spaces $F_k(n, rn)$ for a fixed k and r , and a sequence of random formulas φ_n drawn from the sample spaces, then as $n \rightarrow \infty$ it is almost surely the case that if there exists a σ_n that satisfies φ_n then there exist l more assignments that satisfy φ_n .*

Proof: We will describe a process, for constructing l more satisfying assignments given one satisfying assignment, which will almost surely succeed. Let σ_n be a satisfying assignment for φ_n . All of the clauses of φ_n are of course satisfied and the probability that a satisfied clause will become unsatisfied with a single bit change in σ_n may be found as the product of three factors. First, the probability that the changed variable occurs in the clause, which is $\frac{k}{n}$. Second, given that it is in the clause the probability that the literal in which it occurs has the value 1 before the change, which is $\frac{1}{2}$. Third, given the previous, the probability that the other literals in the clause all have the value 0, which is $\frac{1}{2^{k-1}}$. So the probability that a satisfied clause becomes unsatisfied is $\frac{k}{2^k n}$. Now, the probability that φ_n is not made unsatisfied by a single bit change in σ_n is the same as the probability that none of the rn clauses in φ_n become unsatisfied, i.e.,

$$P(\varphi_n \text{ does not become unsatisfied by one bit change in } \sigma_n) = \left(1 - \frac{k}{2^k n}\right)^{rn},$$

which as $n \rightarrow \infty$ goes to $e^{-\frac{kr}{2^k}}$. Now then, the probability that no single bit change is “good”, i.e., yields a satisfying assignment, is

$$P(\text{all single bit changes in } \sigma \text{ cause } \varphi \text{ to become unsatisfied}) = \left(1 - e^{-\frac{kr}{2^k}}\right)^n.$$

So the probability that we can find a bit to change that will not cause φ_n to become unsatisfied is

$$P(\text{can find a good bit}) = 1 - \left(1 - e^{-\frac{kr}{2^k}}\right)^n.$$

This tends to one as $n \rightarrow \infty$, so almost surely we can find one.

Now the probability that we can find l different good bits can be calculated as the probability that we can find one and then find another that is different from the one we already found and then

another different from the two we already found, etc.

$$P(\text{can find } l \text{ different good bits}) = \prod_{i=0}^{l-1} \left(1 - \left(1 - e^{-\frac{kn}{2^k}} \right)^{n-i} \right)$$

This is a product of a finite number of factors each of which tends to one as $n \rightarrow \infty$, so it tends to one.

■

Theorem 3.4.3: *For any positive integer c , the asymptotic behavior of the properties k -SAT and $\#\text{THRESHOLDSAT}(\geq c)$ are the same in the sample space $F_k(n, rn)$.*

Proof: Let φ be a random formula from the sample space, then clearly, $\varphi \in \#\text{THRESHOLDSAT}(\geq c)$ implies that $\varphi \in k$ -SAT, so $P(\varphi \in k\text{-SAT}) \rightarrow 0$ implies $P(\varphi \in \#\text{THRESHOLDSAT}(\geq 1)) \rightarrow 0$. And from 3.4.2 it follows that $\varphi \in k$ -SAT almost surely implies that $\varphi \in \#\text{THRESHOLDSAT}(\geq c)$, so we also have that $P(\varphi \in k\text{-SAT}) \rightarrow 1$ implies $P(\varphi \in \#\text{THRESHOLDSAT}(\geq 1)) \rightarrow 1$. ■

3.5 Fractional Threshold

The Mean Free Field model suggests that the behavior for a fractional counting threshold is very different from k -SAT. If the clauses-to-variables ratio is used as the critical parameter, the phase transition degenerates in the limit to the origin. In the past this has been pointed out as evidence there is no phase transition [BDK01]; however, this is not strictly the case as the following analysis shows. It is just that for this problem, the threshold function is not $\Theta(n)$, as it is with both the constant and the exponential counting thresholds, but is $\Theta(1)$. So the number of clauses is the appropriate critical parameter to use rather than the clauses-to-variables ratio.

Definition 3.5.1: The decision problem $\#\text{THRESHOLD-}k\text{-SAT}(\geq p/q)$ asks the question: Given a k -CNF formula φ as input, does φ have a number of satisfying assignments that is greater than or equal to p/q of the possible assignments, where p/q is a positive rational fraction greater than 0 and less than 1? ■

Clearly $\#\text{THRESHOLD-}k\text{-SAT}(\geq p/q)$ is a monotone decreasing property for k -CNF formulas and $\text{UN}\#\text{THRESHOLD-}k\text{-SAT}(\geq p/q)$, its complementary decision function, is a monotone increasing property, since adding more clauses to a formula can possibly reduce but never increase the number of satisfying assignments. It immediately follows from a well known theorem of Bollobás and Thomason [BT87] that these decision problems must have thresholds.

Proposition 3.5.2: *For any fixed number of clauses, m , and any random k -CNF formula, φ , that has exactly m clauses, it is the case that:*

$$\lim_{n \rightarrow \infty} \frac{\mathcal{N}_\sigma(\varphi)}{2^n} = \left(\frac{2^k - 1}{2^k} \right)^m$$

Proof: For a fixed m there are a fixed number of places where variables may occur in φ , namely km of them. As $n \rightarrow \infty$, the probability that each of the places is occupied by a different variable from all the others in the formula goes to one, so the satisfiability of any clause is independent of any other clause's satisfiability. Now, for each of the m clauses there are $2^k - 1$ satisfying assignments for its k variables and for the variables that do not occur in any of the clauses there are 2^{n-km} satisfying assignments, so $(2^k - 1)^m 2^{n-km}$ of the 2^n possible assignments of the n variables are satisfying. ■

Proposition 3.5.3: *$f(n) = 1$ is a threshold function for $\text{UN}\#\text{THRESHOLD-}k\text{-SAT}(\geq p/q)$ for all $1 > p/q > 0$.*

Proof: This follows immediately from 3.5.2 and the definition of a threshold function. As $m(n) \rightarrow \infty$ the fraction of satisfying assignments for almost all formulas goes to zero, thus, less than any fixed p/q . On the other hand, as $m(n) \rightarrow 0$ the fraction of satisfying assignments for all formulas goes to one, thus greater than any fixed p/q . ■

Proposition 3.5.4: *The property $\text{UN}\#\text{THRESHOLD-}k\text{-SAT}(\geq p/q)$ has a coarse threshold.*

Proof: Clearly,

$$\lim_{n \rightarrow \infty} \frac{\mu_{1-\epsilon}(m) - \mu_{\epsilon}(m)}{\mu_{\frac{1}{2}}(m)} \geq \frac{1}{\mu_{\frac{1}{2}}(m)}$$

■

The coarse threshold for UN#THRESHOLD- k -SAT($\geq p/q$) also follows from the fact that the set,

$$\{\varphi \mid \text{length of } \varphi = m^* \text{ where } m^* \text{ is the smallest } m \text{ such that } \left(\frac{2^k - 1}{2^k}\right)^m < p/q \},$$

can be used to generate a set that well approximates the property UN#THRESHOLD- k -SAT($\geq p/q$) and clearly its minimal elements are of bounded size, namely, m^* .

3.6 Exponential Threshold

Definition 3.6.1: The decision problem #THRESHOLDSAT($\geq 2^i$) asks the question: Given a CNF φ , and a positive integer i , is the number of satisfying assignments for φ at least 2^i ? ■

Theorem 3.6.2: *The decision problem #THRESHOLDSAT($\geq 2^i$) is PP-complete.*

Proof: We will reduce MAJSAT to #THRESHOLDSAT($\geq 2^i$). Let φ be an instance for MAJSAT. Define, φ' , the transformed instance for #THRESHOLDSAT($\geq 2^i$) as identical to φ and set $i = n - 1$, where n is the number of variables in φ . Note that, n can be determined from examination of φ in polynomial time and the identity transformation is trivially parsimonious and polynomial time. Clearly MAJSAT for φ is YES, iff, #THRESHOLDSAT($\geq 2^i$) is YES for φ' . ■

Definition 3.6.3: The decision problem #THRESHOLDSAT($\geq 2^{\frac{n}{t}}$) asks the question: Given a CNF φ and a positive integer t , is the number of satisfying assignments for φ at least $2^{n/t}$, where n is the number of variables in φ ? ■

Theorem 3.6.4: *The decision problem #THRESHOLDSAT($\geq 2^{\frac{n}{t}}$) is PP-complete.*

Proof: For concreteness we will reduce #THRESHOLDSAT($\geq 2^i$) to #THRESHOLDSAT($\geq 2^{\frac{n}{2}}$).

The case for $t > 2$ clearly follows from a similar argument. Let φ be a CNF over a space of m variables. φ' a parsimonious polynomial time transformation of φ is formed as follows. Its variable space consists of $n = 2m$ variables, the m variables associated with φ , say, $x_1, x_2, x_3, \dots, x_m$, plus m new ones, say, $y_1, y_2, y_3, \dots, y_m$. φ' is set equal to $\varphi \wedge y_1 \wedge y_2 \wedge y_3 \wedge \dots \wedge y_m$. Note that for φ' to be *true*, it must be that $y_1 = y_2 = y_3 = \dots = y_m = \text{true}$. Now the $m - i$ variables $y_{i+1}, y_{i+2}, y_{i+3}, \dots, y_m$ do not occur in φ' , so $|\varphi'| = |\varphi|2^{m-i}$. From this it follows that $|\varphi| \geq 2^i$, iff, $|\varphi'| \geq 2^m = 2^{n/2}$. ■

Theorem 3.6.5: *The decision problem #THRESHOLD-3-SAT($\geq 2^{\frac{n}{3}}$) is PP-complete.*

Proof: We will reduce #THRESHOLDSAT($\geq 2^{\frac{n}{3}}$) to #THRESHOLD-3-SAT($\geq 2^{\frac{n}{3}}$). It is clear that all that is required is to provide a parsimonious and polynomial time transformation for an arbitrary CNF, say φ to a 3CNF, say φ' . Valiant has shown such a transformation must exist between any # P complete problems such as these. For concreteness we give a specific transformation.

First for each k -literal disjunct in φ with $k > 3$ replace the clause, say $(l_1 \vee l_2 \vee \dots \vee l_k)$ with the conjunction of $k - 1$ new clauses:

$$\begin{aligned} y_1 &\wedge (y_1 \leftrightarrow l_1 \vee y_2) \\ &\wedge (y_2 \leftrightarrow l_2 \vee y_3) \\ &\vdots \\ &\wedge (y_{k-1} \leftrightarrow l_{k-1} \vee l_k) \end{aligned}$$

This process introduces $k - 1$ new variables, namely the y s. Each of the new multi-variable clauses contain exactly three variables. They are each replaced by an equivalent four clause 3CNF based on

the following scheme:

$$\begin{aligned}
 (a \leftrightarrow b \vee c) &\equiv (a \vee b \vee \neg c) \\
 &\wedge (a \vee \neg b \vee c) \\
 &\wedge (a \vee \neg b \vee \neg c) \\
 &\wedge (\neg a \vee b \vee c)
 \end{aligned}$$

Now φ' is equivalent to φ and is a CNF formula, all of whose disjuncts consist of not more than 3 literals. Next replace the 1-literal clauses by 3CNF formulas which have the property that there is only one satisfying assignment of their variables and the 1-literal clause is satisfiable iff the 3CNF replacement is satisfiable. The following scheme fits this requirement. For each 1-literal clause, say (l_1) , introduce two new variables, say y_1 and y_2 , and replace the clause with:

$$\begin{aligned}
 &(l_1 \vee y_1 \vee y_2) \\
 &\wedge (l_1 \vee y_1 \vee \neg y_2) \\
 &\wedge (l_1 \vee \neg y_1 \vee y_2) \\
 &\wedge (l_1 \vee \neg y_1 \vee \neg y_2) \\
 &\wedge (\neg l_1 \vee y_1 \vee y_2) \\
 &\wedge (\neg l_1 \vee y_1 \vee \neg y_2) \\
 &\wedge (\neg l_1 \vee \neg y_1 \vee y_2)
 \end{aligned}$$

Lastly, replace the 2-literal clauses by 3CNF formulas which have the property that there are only three satisfying assignments of their variables and the 2-literal clause is satisfiable iff the 3CNF replacement is satisfiable. The following scheme fits this requirement. For each 2-literal clause, say

$(l_1 \vee l_2)$, introduce one new variable, say y_1 , and replace the clause with:

$$\begin{aligned}
 & (l_1 \vee l_2 \vee y_1) \\
 \wedge & (l_1 \vee l_2 \vee \neg y_1) \\
 \wedge & (l_1 \vee \neg l_2 \vee y_1) \\
 \wedge & (\neg l_1 \vee l_2 \vee \neg y_1) \\
 \wedge & (\neg l_1 \vee \neg l_2 \vee y_1)
 \end{aligned}$$

■

3.6.1 Upper Bound

The upper bound for #THRESHOLD- k -SAT ($\geq 2^{\alpha n}$) can be found by using indicator functions, expectation and Markov's inequality as done in our paper [BDK01] and similar to the steps in theorem 2.2.2. Here, however, we will use a direct counting method which gives a better intuitive background for this basic result.

We will use the following simple lemma:

Lemma 3.6.6: *Given, N balls, the maximum number of urns that can be filled with t or more of them is $\lfloor N/t \rfloor$.*

Proof: Clearly, $\lfloor N/t \rfloor$ urns can be filled with t balls. Now, toward a contradiction, assume that more urns can be filled, i.e., $\lfloor N/t \rfloor + j$, where j is an integer greater than or equal to 1. With this assumption the total number of balls in the urns must equal $N + tj > N$, an immediate contradiction. ■

Theorem 3.6.7: $\frac{1-\alpha}{k-1g(2^k-1)}$ is an upper bound for the critical r of the $2^{\alpha n}$ model.

Proof:

Consider table 3.1. It represents a collection of the truth tables for the formulas in $\mathcal{F}_{n,m}^k$ where each cell value, $t_{i,j}$, equals 1 if the i -th formula is satisfied by the j -th assignment and otherwise

Table 3.1: Table of Truth Tables.

	σ_1	σ_2	\dots	σ_j	\dots	σ_{2^n}	
F_1	$t_{1,1}$	$t_{1,2}$	\dots	$t_{1,j}$	\dots	$t_{1,2^n}$	$\sum_j t_{1,j}$
F_2	$t_{2,1}$	$t_{2,2}$	\dots	$t_{2,j}$	\dots	$t_{2,2^n}$	$\sum_j t_{2,j}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
F_i	$t_{i,1}$	$t_{i,2}$	\dots	$t_{i,j}$	\dots	$t_{i,2^n}$	$\sum_j t_{i,j}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$F_{\binom{n}{k}2^k}^m$	$t_{\binom{n}{k}2^k,1}^m$	$t_{\binom{n}{k}2^k,2}^m$	\dots	$t_{\binom{n}{k}2^k,j}^m$	\dots	$t_{\binom{n}{k}2^k,2^n}^m$	$\sum_j t_{\binom{n}{k}2^k,j}^m$
	$\sum_i t_{i,1}$	$\sum_i t_{i,2}$	\dots	$\sum_i t_{i,j}$	\dots	$\sum_i t_{i,2^n}$	$\sum_{i,j} t_{i,j}$

equals 0. So each row corresponds to one particular formula and the row sum is the number of satisfying assignments for that particular formula. Our experiment can be thought of as selecting one of the rows randomly and determining the relationship of its sum to the threshold. Let X be the random variable representing the row sum in this experiment. Then

$$P(X \geq 2^{\alpha n}) = \frac{\text{number of rows with sums greater than or equal to } 2^{\alpha n}}{\binom{n}{k}2^k}^m$$

Now if we knew the total number of 1's in the table we could use lemma 3.6.6 to put a bound on the numerator in the above fraction. This total can be found as the sum of the column sums.

Each column corresponds to one particular assignment and the column sum is the number of formulas that particular assignment makes true. Consider any such assignment, there are exactly $\binom{n}{k}$ disjunct types in \mathcal{C}_n^k that the assignment makes false, one for each combination of k literals in the assignment. So there are $\binom{n}{k}2^k - \binom{n}{k}$ disjunct types that the assignment makes true. Since all the clauses in a true formula must be true, it follows that there are $((2^k - 1)\binom{n}{k})^m$ true formulas for the

assignment. Since this was an arbitrary assignment, this number of true formulas must be the same for all assignments. In other words, all of the column sums are the same. And since there are 2^n columns we have for the total number of 1's in the table

$$\begin{aligned}\sum_j \sum_i t_{i,j} &= \sum_j \left((2^k - 1) \binom{n}{k} \right)^m \\ &= 2^n \left((2^k - 1) \binom{n}{k} \right)^m\end{aligned}$$

and from lemma 3.6.6 the maximum number of rows that equal or exceed the $2^{\alpha n}$ threshold is

$$\frac{2^n \left((2^k - 1) \binom{n}{k} \right)^m}{2^{\alpha n}}$$

and from this it follows that

$$\begin{aligned}\mathbb{P}\{X \geq 2^{\alpha n}\} &\leq \frac{1}{\left(2^k \binom{n}{k}\right)^m} \frac{2^n \left((2^k - 1) \binom{n}{k} \right)^m}{2^{\alpha n}} \\ &\leq \frac{2^n \left(\frac{2^k - 1}{2^k} \right)^m}{2^{\alpha n}} \\ &\leq 2^{(1-\alpha)n} \left(\frac{2^k - 1}{2^k} \right)^m \\ &\leq \left(2^{1-\alpha} \left(\frac{2^k - 1}{2^k} \right)^r \right)^n\end{aligned}$$

Now if a critical r exists, then this probability will be zero as n goes to infinity for all values of r above it. So we look for a range of values of r that are sufficient to make the above expression tend to zero as n goes to infinity. This will be the case if

$$2^{1-\alpha} \left(\frac{2^k - 1}{2^k} \right)^r < 1$$

If r^* is such that

$$2^{1-\alpha} \left(\frac{2^k - 1}{2^k} \right)^{r^*} = 1$$

then for all $r > r^*$ the condition is satisfied and it follows that r^* would be an upper bound for the critical r .

$$\begin{aligned} \left(\frac{2^k - 1}{2^k}\right)^{r^*} &= 2^{\alpha-1} \\ \left(\frac{2^k}{2^k - 1}\right)^{r^*} &= 2^{1-\alpha} \\ r^* \lg\left(\frac{2^k}{2^k - 1}\right) &= 1 - \alpha \\ r^* &= \frac{1 - \alpha}{k - \lg(2^k - 1)} \end{aligned}$$

■

Corollary 3.6.8: *The critical r for the Mean Free Field model is an upper bound for the real model.*

3.6.2 Lower Bound

A lower bound for the critical ratio can be established by using the known analytical result for the the critical ratio of 2-SAT and the concept of a covering partial assignment.

Definition 3.6.9: A partial assignment covers a clause if it is sufficient to force the clause to be true regardless of any possible assignment of the remaining variables. A partial assignment covers a formula, if it covers all of the clauses in the formula. ■

For example, if a partial assignment makes any one of the variables in a clause true, then the clause will be true regardless of the truth values of the variables not fixed by the assignment. Likewise if a partial assignment makes at least one of the variables in each clause of a formula true, then the formula must be true. Note, that if a partial assignment assigns j of n possible variables and covers a formula φ then there are at least 2^{n-j} satisfying assignments for φ , the total number of different ways the unassigned variables could be assigned.

Theorem 3.6.10: $1 - \alpha$ is a lower bound for the critical r^* of the $2^{\alpha n}$ model.

Proof: Given any formula $\varphi \in \mathcal{F}_{n,r < 1-\alpha}^k$ we will find a partial assignment π that covers φ and that has $\leq n(1 - \alpha)$ variables. From this it immediately follows that φ must have at least $2^{\alpha n}$ satisfying assignments.

Since the r for φ is less than $1 - \alpha$ it is certainly less than 1 and the number of clauses in φ must be less than $n(1 - \alpha)$.

For 2-SAT it has been analytically proven [Goe96b, CR92b, Fer92b] that there is a phase transition and it occurs at $r^* = 1$. Now take any φ from $\mathcal{F}_{n,r < 1-\alpha}^k$. Claim: as $n \rightarrow \infty$, it must have at least one satisfying assignment. Let φ' be a 2CNF formula formed from φ by removing the last variable in each of its clauses. This process does not reveal any information about the values of the remaining variables in φ and by the *principal of deferred decision* [Knu76] φ' must be a random 2CNF formula given that φ is a random 3CNF formula. Now if φ' has a satisfying assignment, φ must also have one. Since $r < 1 - \alpha < 1$ as $n \rightarrow \infty$, φ' does have at least one satisfying assignment.

Let σ be a satisfying assignment for φ . We can build a partial assignment π with $\leq n(1 - \alpha)$ variables that covers φ by choosing for each of the $\leq n(1 - \alpha)$ clauses in φ some literal in the clause that is true under σ , and setting π to also make that literal true. ■

3.6.3 Clustering of Number of Satisfying Assignments

The assumption that #THRESHOLD- k -SAT($\geq 2^{\alpha n}$) has a phase transition leads to a remarkable result. The number of satisfying assignments for a random formula clusters around a value determined by the clause-to-variable ratio.

Conjecture 3.6.11: For every integer $k \geq 3$ and every real $\alpha > 0$, there is a positive real number $r_{k,\alpha}$ such that:

- If $r < r_{k,\alpha}$, then $\lim_{n \rightarrow \infty} \mathbb{P}(X_k^{n,r} \geq 2^{\alpha n}) = 1$.
- If $r > r_{k,\alpha}$, then $\lim_{n \rightarrow \infty} \mathbb{P}(X_k^{n,r} \geq 2^{\alpha n}) = 0$.

If this conjecture is true, then there exists a function, $r_k : (0, 1) \rightarrow \mathbb{R}^+$ with $r_k(\alpha)$ equal to the $r_{k,\alpha}$ asserted in the conjecture. This function is clearly monotonic decreasing and therefore has an inverse, say r_k^{-1} .

Now for any $\epsilon > 0$, consider r such that $r - \epsilon < r < r + \epsilon$. It follows immediately from the conjecture that it must be the case that

$$\lim_{n \rightarrow \infty} \mathbb{P}(2^{r_k^{-1}(r-\epsilon)n} < X_k^{n,r} < 2^{r_k^{-1}(r+\epsilon)n}) = 1$$

3.6.4 Tightness of Upper Bound

For the 3SAT phase transition, the upper bound determined by the first moment argument is very loose, ≈ 5.19 versus ≈ 4.2 . However, for the $2^{\alpha n}$ counting-threshold phase transitions the combinatorial argument above gives much tighter bounds with respect to experiment as can be seen in table 3.7.3. Since the combinatorial argument is equivalent to the first moment technique it is interesting to ask why it produces tighter results in these cases.

The following urn and ball model gives some insight into what is happening.

Let $U = \{u_1, u_2, u_3, \dots, u_n\}$ be a finite set of N urns. Consider distributions of n balls among the urns in U and a counting-threshold number t . Let $\hat{U} = \{u \in U \mid |u| \geq t\}$ and $\dot{U} = \{u \in U \mid |u| > 0\}$, and v be a randomly selected urn from U . From lemma 3.6.6 we have that $|\hat{U}| \leq n/t$. Now consider what happens depending on how t compares to the average number of balls in occupied urns, $\text{aveO} = n/|\dot{U}|$. There are three cases:

- $t < \text{aveO}$. There can not be more urns containing t or more balls than there are urns with 1 or more balls. So if $t \ll n/|\dot{U}|$, then the Markov bound n/t is very loose. See figure 3.1
- $t = \text{aveO}$. If all the occupied urns have the same number of balls, the Markov bound would be achieved, so in this case it is a least upper bound.
- $t > \text{aveO}$. In this case $\frac{n}{t}$ is less than the number of occupied urns, so not all of the n balls can

contribute to filling urns up to the threshold value, t . The most balls that could be unavailable is n/aveO , so the least upper bound is greater than $\frac{n}{t}(1 - 1/\text{aveO})$. So if aveO is very large, the Markov bound is essentially the least upper bound. See figure 3.2.

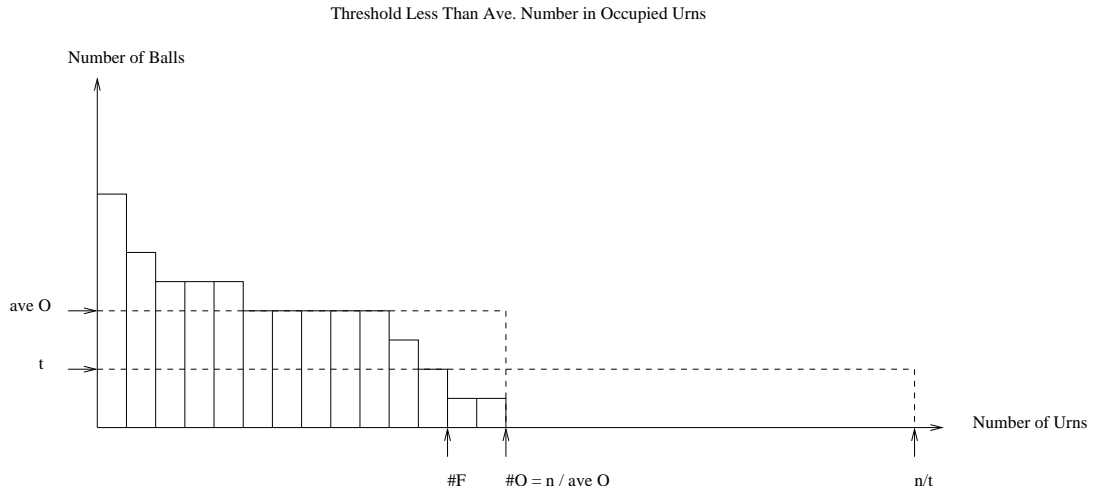


Figure 3.1: Threshold Less Than

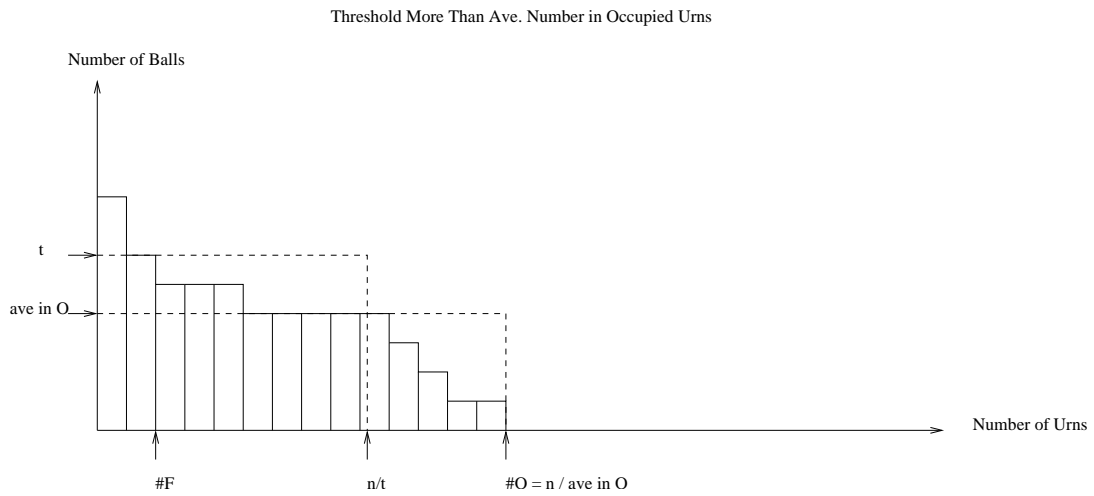


Figure 3.2: Threshold More Than

With the formula spaces, the different possible formulas can be thought of as the urns and a satisfying assignment as a ball. Now we know for $\#\text{THRESHOLD-}k\text{-SAT}(\geq 2^{\alpha n})$ the expected

number of satisfying assignments for a random formula from $F_k(n, rn)$ is

$$2^n \left(\frac{2^k - 1}{2^k} \right)^{rn}$$

and for r less than $r^*(n)$, the sharp transition point for *ksat*, that almost surely random formulas have some satisfying assignment. Therefore, the above expression is also the expected number of satisfying assignments for a satisfied formula from $F_k(n, rn)$ if $r < r^*(n)$ or in other words aveO. So, when $r = r_{k,\alpha}$, the $2^{\alpha n}$ counting threshold is equal to aveO and when $r_{k,\alpha} < r < r^*(n)$, the $2^{\alpha n}$ counting threshold is larger than aveO. Thus, the Markov bound is tight for the $2^{\alpha n}$ counting threshold. Note, that for k -SAT the threshold value is 1 and in the region from $r^*(n)$, ≈ 4.2 , to the Markov bound for k -SAT, ≈ 5.19 , the expected number of satisfying assignments for a satisfied formula grows without bound, so the threshold of 1 becomes arbitrarily small compared to it and the Markov bound is quite loose.

3.7 Experiments

3.7.1 Method

We implemented threshold counting algorithms and ran experiments for random 3CNF-formulas with 10, 20, 30 and 40 variables, For each space, individual runs were made for thresholds of $2^{\alpha n}$ with α varying from 0.1 to 0.9 in 0.1 steps. For each sample point, (n, m, α) , we generated 1200 random 3CNF-formulas with sizes from 1 to 200 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. We measured the costs to process the 1200 random formulas and counted the number of formulas that were “yes” instances.

3.7.2 Algorithms

We modified Birnbaum and Lozinskii's basic CDP (Counting Davis-Putnam Procedure) to make a TCDP (Threshold Counting Davis-Putnam Procedure).

The basic CDP is a recursive function $\text{CDP}(\varphi, n)$ where φ is a Boolean CNF formula that contains no clause which is a tautology and n is the number of variables in the space considered. It is similar to the DPLL (Davis-Putnam-Logeman-Loveman) algorithm, but much simpler.

- if φ is empty, return 2^n
- if φ contains an empty clause, return 0
- if φ contains a unit clause $\{u\}$

return $\text{CDP}(\varphi|_{u \leftarrow 1}, n - 1)$

- otherwise choose any variable v in φ

return $\text{CDP}(\varphi|_{v \leftarrow 1}, n - 1) + \text{CDP}(\varphi|_{v \leftarrow 0}, n - 1)$

Where, since φ is a CNF, the restrictions $\varphi|_{x \leftarrow 1}$ and $\varphi|_{x \leftarrow 0}$ are as follows:

- $\varphi|_{x \leftarrow 1}$ contains all clauses in φ that do not contain x ; with the literal $\neg x$ removed if present
- $\varphi|_{x \leftarrow 0}$ contains all clauses in φ that do not contain $\neg x$; with the literal x removed if present

The basic-TCDP is the CDP with the added feature that a running count of the number of satisfying assignments determined at any time is maintained. When the count equals or exceeds the counting threshold the algorithm terminates. It is a recursive function $\text{bTCDP}(\varphi, n, t, \text{LB})$ where φ is a Boolean CNF formula that contains no clause which is a tautology, n is the number of variables in the space considered, t is the counting threshold value being tested, LB is the currently known lower bound on the count. TCDP returns 1 if it detects that the count will equal or exceed t , returns 0 if the count is completed and is less than t . Initially LB is set to 0.

- if φ is empty
 - $LB \leftarrow LB + 2^n$
 - if $LB \geq t$
 - then return 1
 - else return 0
- if φ contains an empty clause, return 0
- if φ contains a unit clause $\{u\}$
 - return $\text{bTCDP}(\varphi|_{u \leftarrow 1}, n - 1, t, LB)$
- otherwise choose any variable v in φ
 - temp = $\text{bTCDP}(\varphi|_{v \leftarrow 1}, n - 1, t, LB)$
 - if temp = 1, then return temp
 - else return $\text{bTCDP}(\varphi|_{v \leftarrow 0}, n - 1, t, LB)$

We also implemented a symmetric-TCDP in which both an upper bound and a lower bound on the count are maintained. Initially, of course, the upper bound is 2^n and the lower bound is 0. The lower bound at any time is the accumulated count so far. Termination occurs whenever the lower bound (i.e. the accumulated count) equals or exceeds the counting threshold. This is the same as in the basic-TCDP. With the symmetric version, however, termination also occurs whenever the upper bound becomes lower than the counting threshold in which case we know that additional counting cannot result in the counting threshold being met.

The symmetric-TCDP is a recursive function $\text{sTCDP}(\varphi, n, t, LB, UB)$ where φ is a Boolean CNF formula that contains no clause which is a tautology, n is the number of variables in the space considered, t is the counting threshold value being tested, LB is the currently known lower bound on the count, and UB is the currently known upper bound on the count. The sTCDP function returns 1 if it detects that the count will equal or exceed t , returns 0 if the count is completed and is less than t ,

and returns -1 if it detects that the count will not be able to reach t or more. Initially LB and UB are respectively set to 0 and 2^n .

- if φ is empty

$$\text{LB} \leftarrow \text{LB} + 2^n$$

$$\text{if } \text{LB} \geq t$$

then return 1

else return 0

- if φ contains an empty clause

$$\text{UB} \leftarrow \text{UB} - 2^n$$

$$\text{if } \text{UB} < t$$

then return -1

else return 0

- if φ contains a unit clause $\{u\}$

$$\text{UB} \leftarrow \text{UB} - 2^{n-1}$$

$$\text{if } \text{UB} < t$$

then return -1

else return $\text{sTCDP}(\varphi|_{u \leftarrow 1}, n-1, t, \text{LB}, \text{UB})$

- otherwise choose any variable v in φ

$$\text{temp} = \text{sTCDP}(\varphi|_{v \leftarrow 1}, n-1, t, \text{LB}, \text{UB})$$

$$\text{if } \text{temp} = 1 \text{ or } -1$$

return temp

else return $\text{sTCDP}(\varphi|_{v \leftarrow 0}, n-1, t, \text{LB}, \text{UB})$

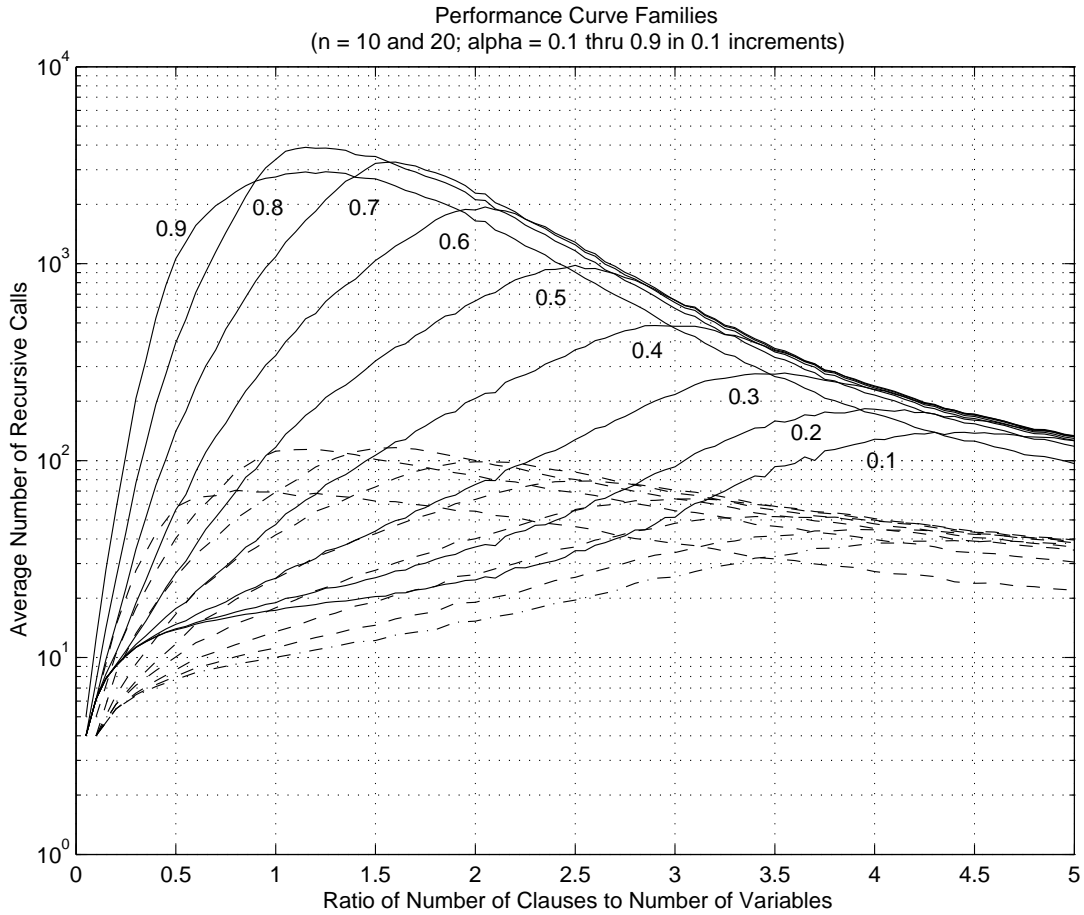


Figure 3.3: Average Search Cost Graphs for 10 and 20 Variables

3.7.3 Results

Experiments were run for random 3CNF-formulas with 10, 20, 30 and 40 variables by implementing the symmetric TCDP algorithm on a dual 1GHz i686s/4GB memory/Linux 2.4.2-2smp workstation with C and the GNU Multiple Precision package. For each space, individual runs were made for thresholds of $2^{\alpha n}$ with α varying from 0.1 to 0.9 in 0.1 steps.

The results are depicted in Figures 3.3, 3.4 and 3.5. In these figures the horizontal axis is the ratio of the number of clauses to the number of variables in the space. The ranges of formula sizes represented in the graphs are 1 to 50, 1 to 100, 1 to 150, and 1 to 200 for the 10, 20, 30 and 40 variable

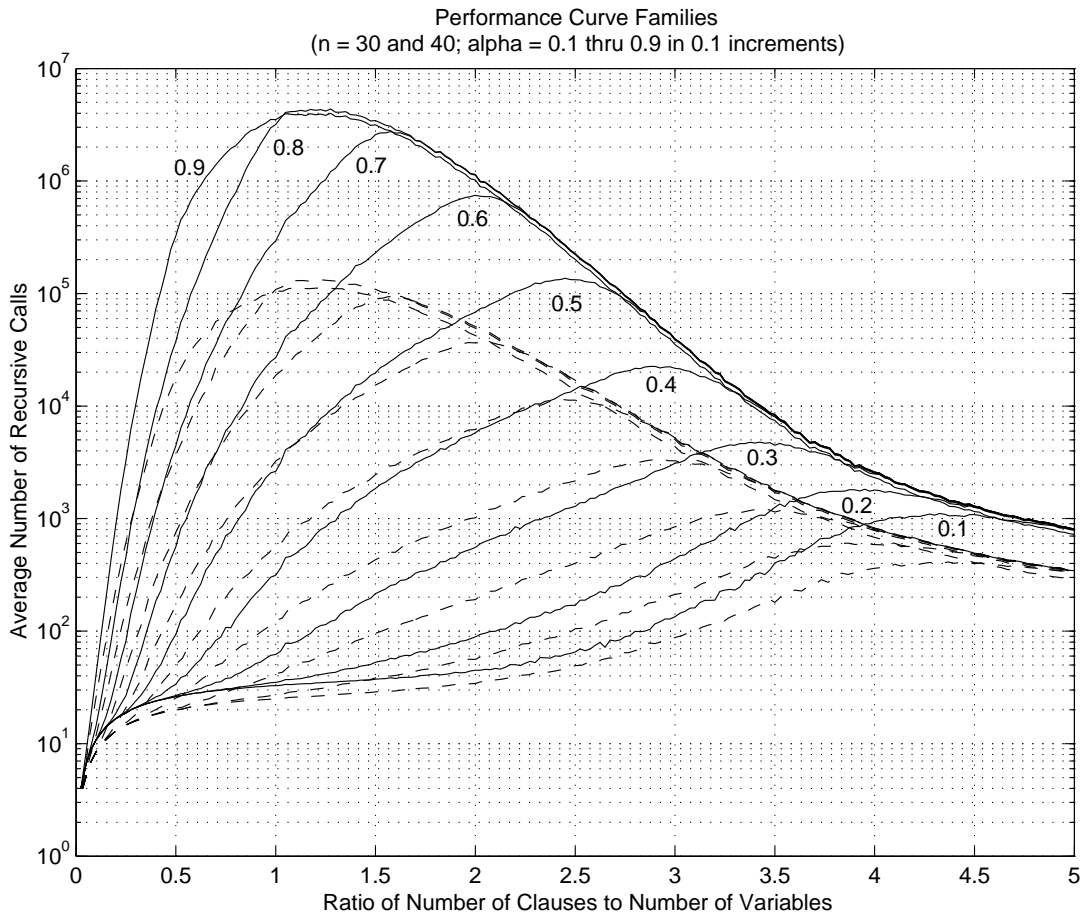


Figure 3.4: Average Search Cost Graphs for 30 and 40 Variables

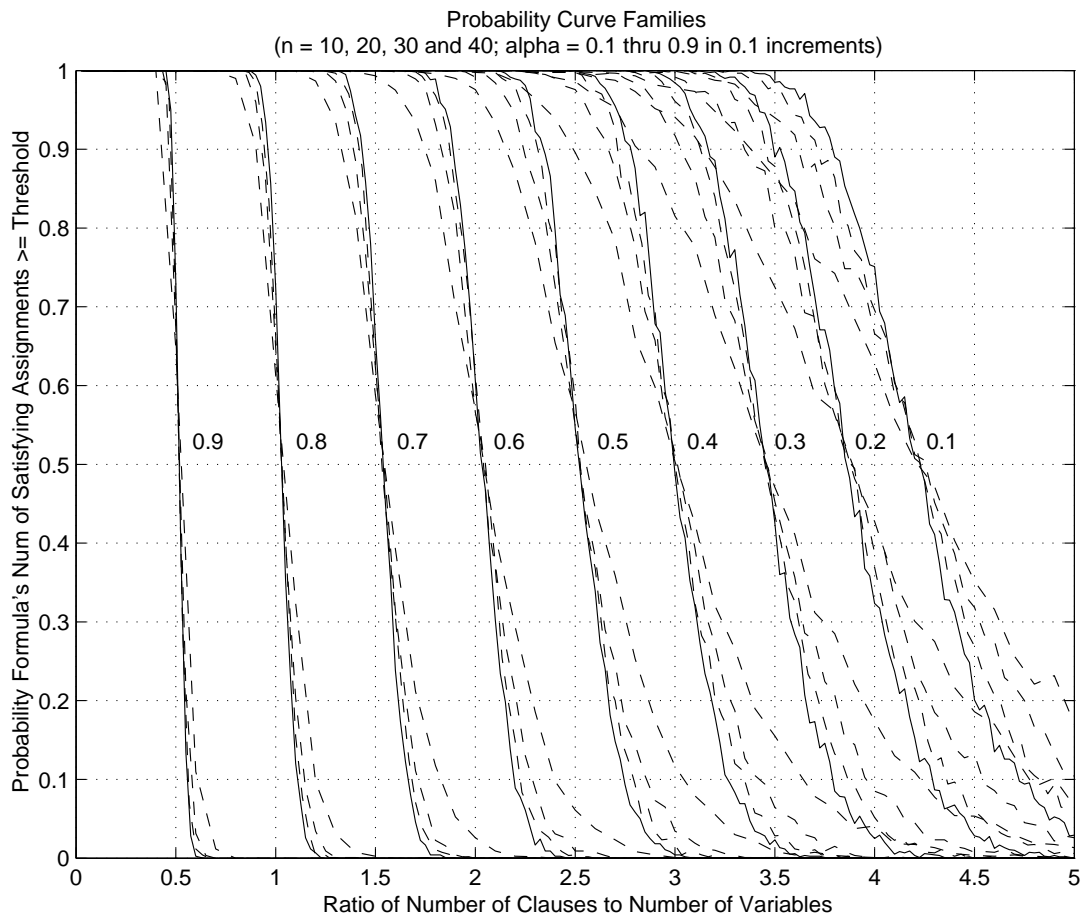


Figure 3.5: Probability Phase Transition Graphs

Table 3.2: Markov Upper Bound vs. Estimate for r^*

α	MUB $r^*(\alpha)$	Estimate $r^*(\alpha)$	Transition Window	Peak Cost ($n = 40$)
0.1	4.6718	4.19	[3.775, 4.725]	4.5-
0.2	4.1527	3.84	[3.475, 4.300]	4.0-
0.3	3.6336	3.45	[3.150, 3.800]	3.4
0.4	3.1145	3.00	[2.775, 3.275]	3.0-
0.5	2.5954	2.50	[2.325, 2.750]	2.5-
0.6	2.0764	2.02	[1.875, 2.175]	2.0
0.7	1.5573	1.53	[1.425, 1.675]	1.6-
0.8	1.0382	1.02	[0.950, 1.200]	1.2
0.9	0.5191	0.51	[0.475, 0.575]	1.2

spaces respectively.

The probability phase transition graphs in Figure 3.5 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 $2^{\alpha n}$ threshold. For each α , the window in which the probability drops from 1 to 0 becomes narrower and steeper as the number of variables increases. We used finite-size scaling, assuming a power law of the form $\frac{(r-r^*(\alpha))n^\nu}{r^*(\alpha)}$, to obtain estimates for the critical ratio $r^*(\alpha)$ and for the exponent ν . The estimates for the values of $r^*(\alpha)$ are given in the second column of Table 3.2; the respective estimates for the values of ν , as α varies from 0.1 to 0.9 in 0.1 steps are: 0.4968, 0.5112, 0.5899, 0.5849, 0.5800, 0.5021, 0.5021, 0.4931, and 0.6598. When the data were accordingly rescaled, the four curves ($n = 10, 20, 30, 40$) for each α -family collapsed to a single curve, thus providing further evidence for the existence of a phase transition at $r^*(\alpha)$. Table 3.2 shows how the estimates for $r^*(\alpha)$ compare with the upper bounds obtained in theorem 3.6.7.

The average search cost graphs in Figures 3.3 and 3.4 show the average number of recursive calls required by the symmetric TCDP to test each set of 1200 sample formulas. The run-times varied from a few minutes to a few days. Whereas, with the probability curves we had distinct families for each value of α , here we have distinct families corresponding to each value of n . Figure 3.3 depicts the families for $n = 10$ and $n = 20$, while Figure 3.4 depicts the families for $n = 20$ and $n = 40$.

Obviously, the number of recursive calls increases with increasing values of n . The interesting outcome is that for any particular value of n the difficulty varies with both α and r . For a fixed n and fixed α , there is the characteristic “easy-hard-easier” pattern of difficulty for increasing values of r . Moreover, as α increases from 0.1 to 0.9, the peaks move to the left, for example in the 40 variable family, they appear to occur at 4.5–, 4.0–, 3.4, 3.0–, 2.5–, 2.0, 1.6–, 1.2, 1.2.

For every α , we define the *transition window* for $\#3\text{SAT}(\geq 2^{\alpha n})$ to be the interval of ratios r in which the 40 variable curve drops from a probability of 0.9 to 0.1. Note that when $\alpha \leq 0.7$, the ratio r at which the peak cost occurs is well inside the transition window; in fact, it either coincides with or is close to the experimental estimate for $r^*(\alpha)$, as would be expected from the conventional conjecture that the peak cost occurs at or near the phase transition point (see Table 3.2). The state of affairs, however, changes for $\alpha = 0.8$ and $\alpha = 0.9$. Indeed, the $\alpha = 0.8$ curve peaks at 1.2, which is the boundary of the transition window. Even more dramatically, the $\alpha = 0.9$ curve also peaks at 1.2, which is well to the right of the transition window $[0.475, 0.575]$. See Figure 3.6.

Also noteworthy is the fact that the peak cost for $\alpha = 0.8$ and $\alpha = 0.9$ occurs at $r = 1.2$, which is the ratio at which the CDP procedure for $\#3\text{SAT}$ peaks [BL99]. We now provide an analysis of the relationship between the peak cost of CDP and that of the symmetric TCDP or other variants of it.

A naive algorithm to solve threshold counting satisfiability problems, such as $\#3\text{SAT}(\geq 2^{\alpha n})$, is to simply run CDP to find the exact number of satisfying assignments and then compare the result with the threshold count. The most obvious and direct improvement to this naive algorithm is to consider a threshold variant of CDP in which a lower bound on the count is maintained; we will refer to this variant as the *basic TCDP*. The symmetric TCDP considered here is a further refinement of the basic TCDP in which both a lower bound and an upper bound are maintained. In other words, the basic TCDP can be obtained from the symmetric TCDP by disabling the upper bound check.

Threshold variants of CDP, such as the basic TCDP and the symmetric TCDP, only allow for speeding it up. They terminate earlier than CDP does (i.e., before a full count is completed) only

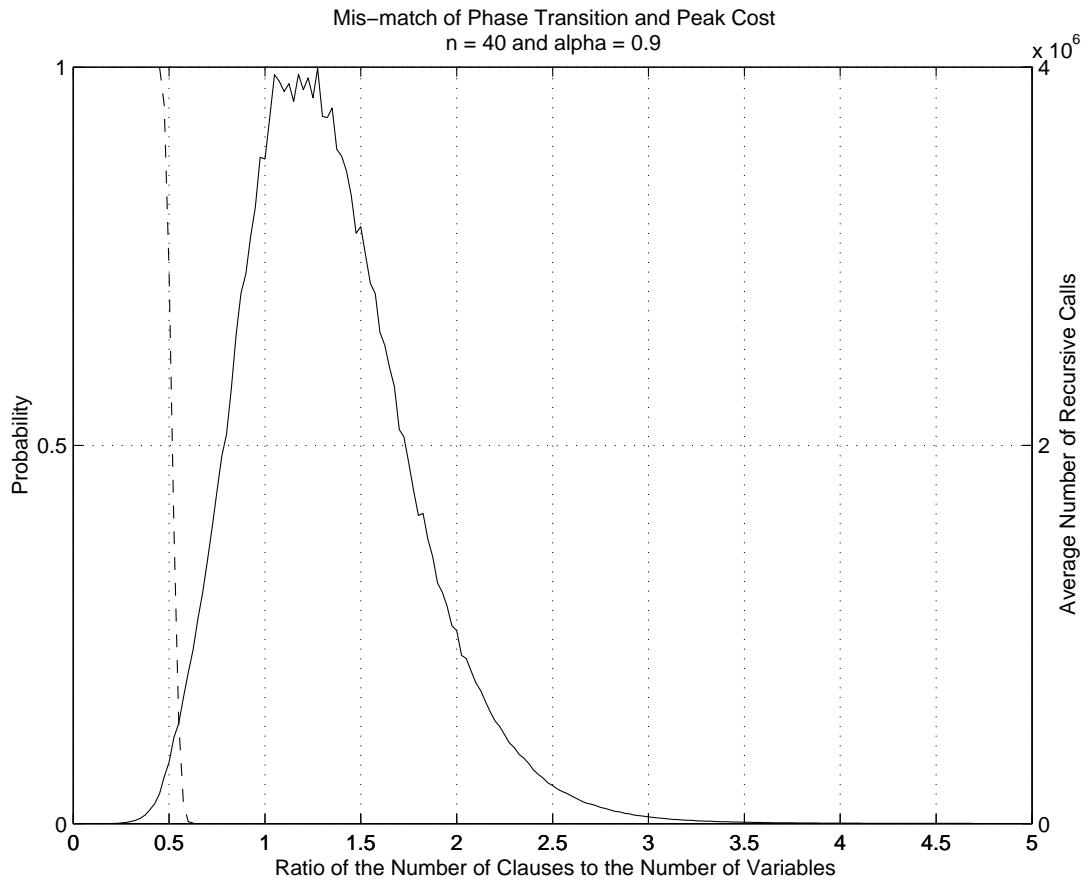


Figure 3.6: Peak Cost Outside Transition Window

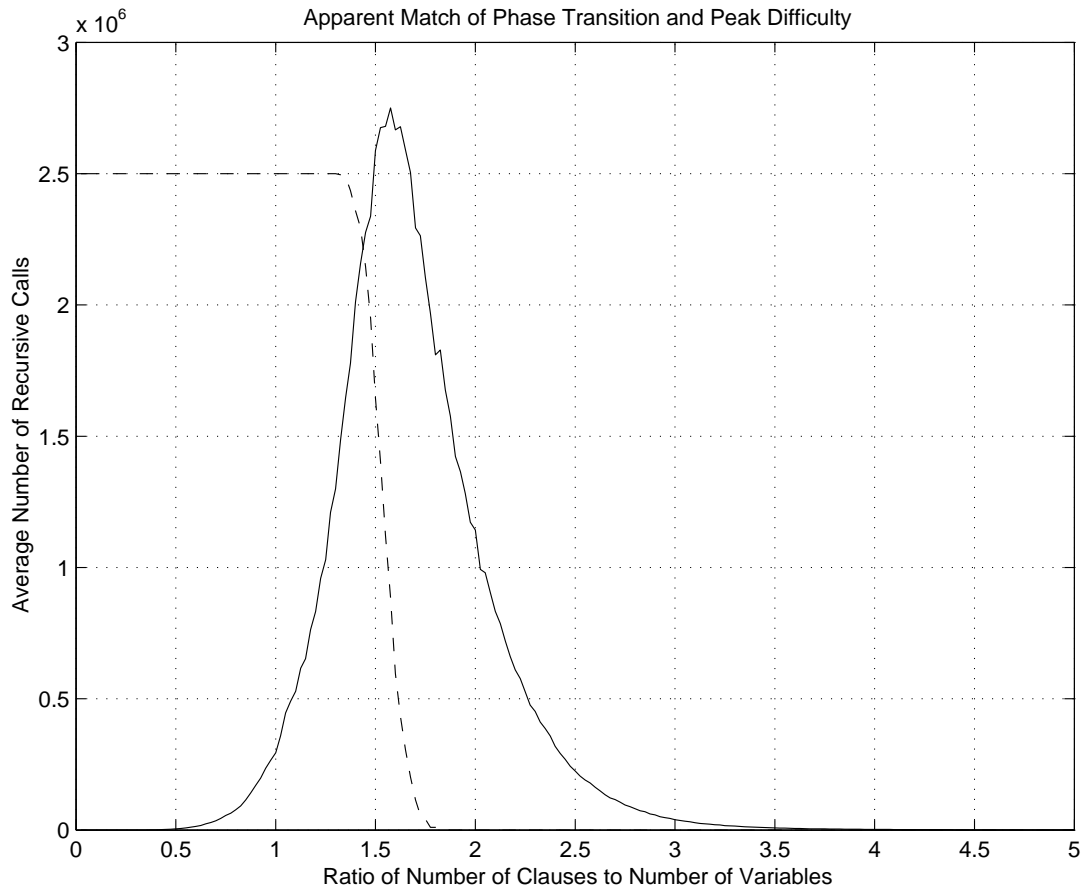


Figure 3.7: Peak Cost Inside Transition Window

when they determine that the threshold is exceeded or that it cannot be exceeded. They will make a full count, however, if this early termination does not occur. Consequently, for every CNF-formula φ , the number of recursive calls of the basic TCDP, or of the symmetric TCDP on φ , required to determine if the number of satisfying assignments is greater than or equal to a threshold value t , cannot exceed the number of CDP recursive calls required to count all of the satisfying assignments. Moreover, if the number of satisfying assignments is less than the threshold value t , then the number of recursive calls required by the symmetric TCDP and the CDP will be equal, except for cases in which the upper bound check is able to cause early termination. By the same token, if the number of satisfying assignments is less than the threshold value t , then the number of recursive calls of the basic TCDP will be equal to those of the CDP.

Now, from the preceding comments and the definition of the critical ratio $r^*(\alpha)$, it follows that as n becomes arbitrarily large, for every $r > r^*(\alpha)$ the difference between the cost curves of the basic TCDP and the CDP will essentially diminish. Moreover, for every $r < r^*(\alpha)$ the cost curves for the basic and the symmetric TCDP will be lower than the cost curve for the CDP. (Even though the existence of the critical ratios $r^*(\alpha)$ has not been established analytically, the above remains true, when r is taken to be respectively bigger or smaller than the upper and lower bounds for $r^*(\alpha)$.) Finally, suppose that the cost curve of the CDP for n variables has a peak at some ratio r_c and consider an α such that the critical value $r^*(\alpha)$ for the phase transition is less than r_c . In this case, the peak cost of the basic TCDP *cannot* occur at $r^*(\alpha)$.

These remarks suggest a qualitative way that we can predict and describe the peak formation in the average cost curves of the basic TCDP. When the critical $r^*(\alpha)$ for a particular α is greater than the ratio r_c value at which CDP peaks, then the average cost curves of the basic TCDP for that α will nearly match the CDP curve for all r values to the right of the phase transition region because the formulas concerned have a very low probability of having enough satisfying assignments to cause the basic TCDP to terminate doing a complete count. As r values move into the phase transition region,

formulas will begin to have enough satisfying assignments to terminate the algorithm early and cause the performance curve to start to break away from the CDP curve. To the left of the phase transition region nearly every formula will have enough satisfying assignments to cause the early termination (also note that there are greater and greater numbers of satisfying assignments as r gets smaller and smaller since smaller r 's correspond to fewer constraints). On the other hand, when the critical $r^*(\alpha)$ for a particular α is less than the r_c value for the peak difficulty of the CDP, then the peak difficulty of the basic TCDP for that α must match the peak for the CDP, since, coming from the right, the break will not occur until $r < r^*(\alpha)$, which means r 's to the left of the CDP peak.

Strictly speaking, the above analysis applies to the basic TCDP algorithm. Our experiments with the symmetric TCDP depicted in Figures 3.3 and 3.4 reveal that the symmetric TCDP basically exhibits a similar behavior except that the curves appear to drop for higher α values. To further corroborate these findings, we ran experiments with the CDP and the basic TCDP. Figure 3.8 shows the results for 20 variable runs; it also includes the curve for CDP, which is an envelope for the curves of the basic TCDP. We note that at its peak the symmetric TCDP requires about 4,000 recursive calls for $\alpha = 0.8$ and about 3,000 recursive calls for $\alpha = 0.9$, while at its peak, the basic TCDP requires about 4,200 recursive calls for $\alpha = 0.8$ and also for $\alpha = 0.9$. We also note that, for $\alpha = 0.9$, the region $r < 0.5$ is the only region of ratios in which the difference in performance between CDP and the basic TCDP is apparent.

Bayardo and Pehoushek [BP00] designed and implemented a different DPLL extension for solving #SAT, called Decomposing Davis-Putnam (DDP), which utilizes connected components in the constraint graph associated with a CNF-formula. Their experiments showed that the DDP performs better than the CDP and that its average peak cost occurs when $r \approx 1.5$. The analysis presented earlier is also applicable to the DDP and its threshold variants, as regards the qualitative relationship between the location of phase transitions for #3SAT($\geq 2^{\alpha n}$), $0 < \alpha < 1$, and search cost of threshold variants of DDP on this family.

In our experiments we report the average rather than the median cost because of its intrinsic relationship of average to expectation. It should be noted that other researchers have considered both median performance and average (mean) performance in phase transition experiments for certain NP-complete problems, and have discovered differences in the behavior of these quantities. In our experiments the median behaves very similarly to the average, even for strikingly different values of α , as can be seen in Figure 3.9. This should be contrasted with the findings of other investigators [HW94, Bak95] concerning differences between the average cost and the median cost for solving GRAPH COLORING, and similar findings [GW94] for 3SAT under the constant probability model.

In conclusion, in this part, we have reported on our discovery and study of the family $\#3SAT(\geq 2^{\alpha n})$, $0 < \alpha < 1$, of PP-complete satisfiability problems, each of which exhibits a phase transition at a different ratio $r^*(\alpha)$ that depends on the parameter α . We also investigated the average peak cost of the symmetric TCDP, a natural threshold counting algorithm for solving instances of these problems. Since the occurrence of the phase transition differs from problem to problem in the family, we have been able to see how the phase transition affects the performance of the algorithm and, in the process, have discovered that peak cost does not always occur at the location of the phase transition nor does it necessarily occur at the cross-over point.

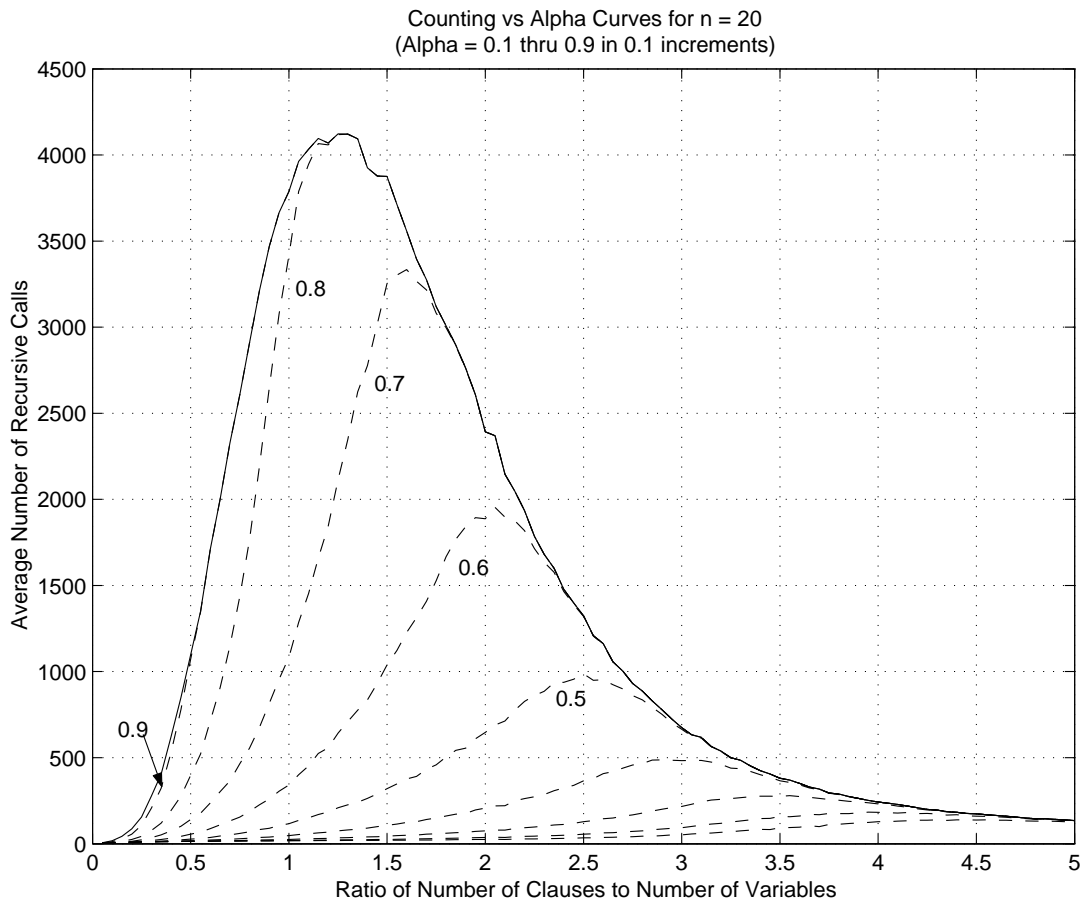
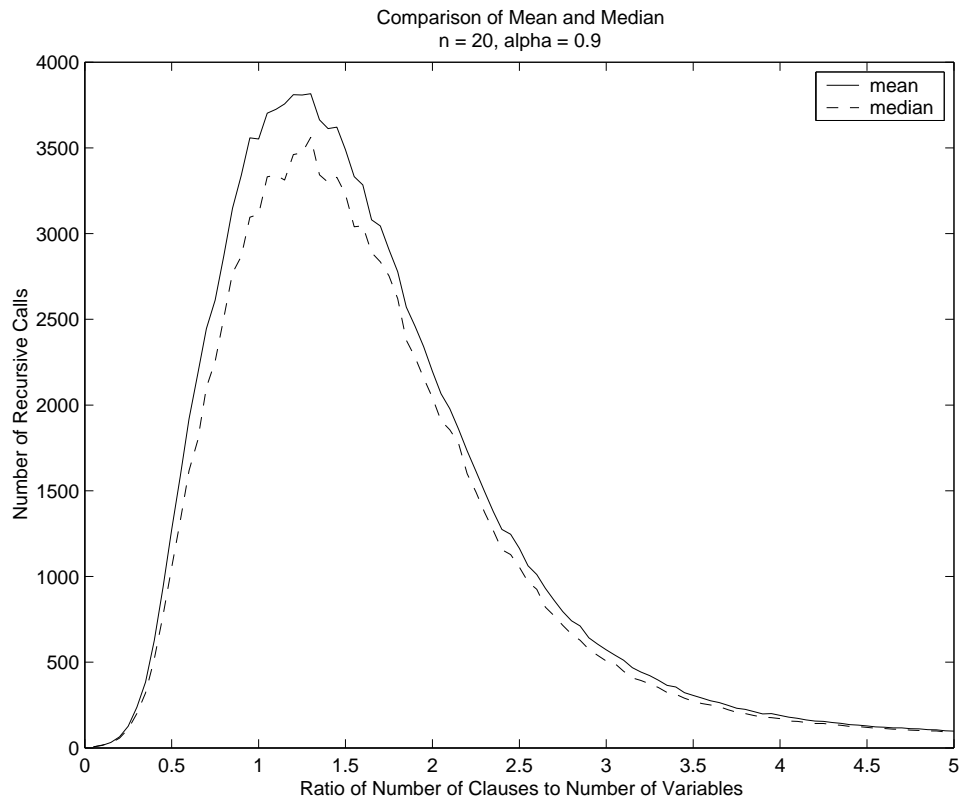
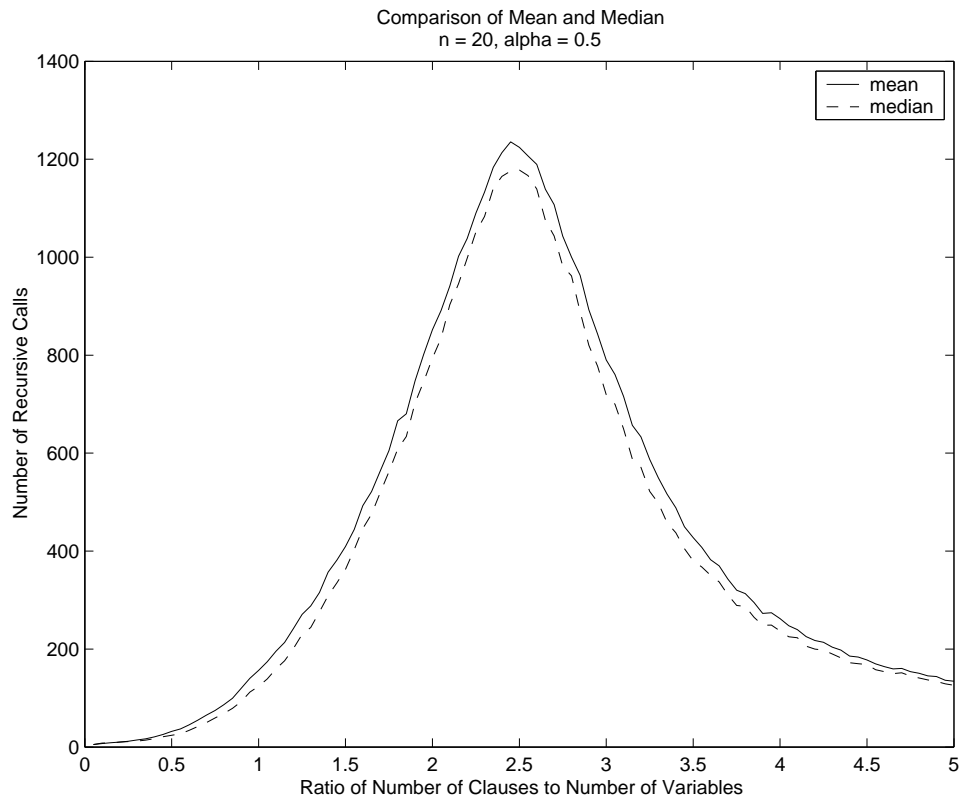


Figure 3.8: CDP vs. Basic TCDP Search Costs



72
Figure 3.9: Mean and Median Track

Part IV

Bounded Satisfiability Problems

Chapter 4

Bounded Satisfiability

4.1 Introduction

Many fundamental algorithmic problems are optimization problems and not mere decision problems. In particular, if a Boolean formula happens to be overconstrained, and thus, unsatisfiable, it is natural to ask for the maximum number of clauses in the formula that can be simultaneously satisfied. By focusing on 3CNF-formulas, we obtain the optimization problem MAX 3SAT: given a 3CNF-formula φ , find the maximum number of clauses of φ that can be simultaneously satisfied. MAX 3SAT is a prototypical constraint optimization problem that is known to play a prominent role in the study of the approximability properties of NP-optimization problems. Indeed, as shown by Papadimitriou and Yannakakis [PY91], MAX SAT is a MAX SNP-complete problem; this means that MAX SAT is a constant-approximable optimization problem and that every NP-optimization problem in the class MAX SNP can be reduced to it via an approximation-preserving polynomial-time reduction [Pap94].

Its importance in complexity and approximability notwithstanding, MAX 3SAT had not been investigated from the perspective of phase transitions until recently. Zhang [Zha01] was the first to investigate phase transitions for MAX 3SAT by carrying out an initial set of experiments for the

family of *bounded satisfiability* problems $3\text{-SAT}(b)$, where b is a non-negative integer. An instance of $3\text{-SAT}(b)$ is a 3CNF-formula φ and the question is: does there exist a truth assignment that violates no more than b clauses of φ ? Equivalently, $3\text{-SAT}(b)$ asks whether the optimum value of MAX SAT on an instance φ is at least $m - B$, where m is the number of clauses of φ . Thus, each $3\text{SAT}(b)$ is a decision problem obtained from MAX SAT by imposing a “quality bound” on the optimum value. The control parameter of each $3\text{-SAT}(b)$ is the ratio of clauses-to-variables, that is, the same control parameter as the one used for 3SAT.

Zhang [Zha01] explored phase transitions for the bounded satisfiability problems $3\text{-SAT}(b)$ by running experiments on random 3CNF-formulas with $n = 25$ variables and for $b = 5, 10, 15, 20$. His findings suggest that there is a series of separated phase transitions corresponding to different quality bounds. The location of each phase transition appears to increase with b ; moreover, the average cost for solving the optimization problem MAX SAT appears to be the envelope of the peak average cost for solving the decision problems $3\text{-SAT}(b)$, as b increases. Zhang [Zha01] did not report on the asymptotic behavior of $3\text{-SAT}(b)$ as the number of variables increases because he ran experiments for just a single value ($n = 25$) of the number of variables. He posed the analysis of phase transitions for $3\text{-SAT}(b)$ as an open problem and, in particular, raised the question of finding the exact location of the phase transitions for these decision problems.

In this section, we report on a systematic investigation of phase transitions for the family of bounded satisfiability problems $3\text{-SAT}(b)$. Our investigation has produced both analytical and experimental results that yield a more complete and, at the same time, rather surprising picture of these phase transitions. As stated above, Zhang’s [Zha01] initial experiments suggest that the location of the phase transition for each $3\text{-SAT}(b)$ increases with b . Here, using the first-moment method, we show analytically that the phase transitions for $3\text{-SAT}(b)$ must occur, if they exist, within a rather narrow region, regardless of the value of b . Moreover, the same behavior is exhibited by the families $k\text{SAT}(b)$ of decision problems underlying the optimization problems MAX $k\text{SAT}$, $k \geq 3$. In particular, the

locations of the phase transitions for $3\text{-SAT}(b)$ are bounded from above by $1/(3 - \lg(7)) \approx 5.19$ and from below by the greatest known lower bound 3.42 for the location of the phase transition for 3SAT. At the experimental front, we investigated the asymptotic behavior of $3\text{-SAT}(b)$ for the values $b = 3, 4$ and 5 by running experiments for several different values of the number n of variables of random 3CNF-formulas ($n = 10, 15, 20, 25, 30, 35$). Our most striking experimental finding is the discovery that, as n increases, the phase transition for each $3\text{-SAT}(b)$ emerges in a manner that is qualitatively different from that for 3SAT.

In the case of 3SAT, the experiments of Selman et al. [SML96] revealed that the family of curves for the probability of satisfiability of a random 3CNF-formula (one curve for each number n of variables) has the property that the curves become progressively steeper, and every two of them intersect at a single point whose abscissa is near the location of the phase transition for 3SAT (see Figure 4.1). In contrast, our experiments reveal that, for each bounded satisfiability problem $3\text{-SAT}(b)$, the family of curves for the probability of a “yes” answer of a random 3CNF-formula (one curve for each number n of variables) become progressively steeper but they do *not* intersect; instead, they are separated by a distance that is getting smaller as n increases. Figures 4.7, 4.9, and 4.11 depict these findings for $3\text{-SAT}(3)$, $3\text{-SAT}(4)$ and $3\text{-SAT}(5)$, respectively. This is a qualitatively different pattern of emergence of a phase transition that does not seem to have been encountered in earlier investigations of phase transitions of other NP-complete problems. This new phenomenon implies that the locations of the phase transitions for the bounded satisfiability problems $3\text{-SAT}(b)$ *cannot* be estimated by a mere visual inspection of the intersection point of the probability curves. Instead, these locations have to be estimated through an application of the finite-size scaling method. Using this method, we obtained the values 4.4, 4.48 and 4.84 as estimates of the location of the phase transition for $3\text{-SAT}(3)$, $3\text{-SAT}(4)$ and $3\text{-SAT}(5)$, respectively.

4.2 Bounded Satisfiability Problems

For every positive integer $k \geq 3$, MAX k SAT is the following optimization problem: given a k -CNF-formula φ , find the maximum number of simultaneously satisfied clauses of φ . The study of phase transitions for an optimization problem begins by focusing on one or more decision problems that underlie that optimization problem. The most natural way to derive a decision problem from an optimization problem is to consider as input both an instance of the optimization problem and an arbitrary quality bound, and to ask whether the optimum value on this instance is bigger (or smaller) than the given quality bound. In the case of MAX k SAT, this gives rise to the following NP-complete decision problem: given a k -CNF-formula φ and an integer c , is there a truth assignment to the variables of φ that satisfies at least c clauses of φ ? It is also possible to derive a family of decision problems that underlie MAX k SAT, and are such that the inputs to each of them are k -CNF-formulas only. For each non-negative integer b , let k SAT(b) be the following *bounded satisfiability* problem: given a k -CNF-formula φ , is there a truth assignment to the variables of φ that violates no more than b clauses of φ ? Clearly, k SAT(0) is k -SAT itself; moreover, it is easy to show that every bounded satisfiability problem k SAT(b), where $k \geq 3$ and $b \geq 1$, is NP-complete. Since the input to k SAT(b) is a k -CNF-formula, phase transitions for the family k SAT(b) can be studied using the ratio of clauses to variables as the control parameter, that is, the same control parameter as the one used in the study of phase transitions for k -SAT.

There have been earlier investigations of phase transitions for other fundamental optimization problems, including the TRAVELING SALESMAN PROBLEM [GW96b] and NUMBER PARTITIONING [GW96a]. It should be noted, however, that for each of these two optimization problems a single decision problem was considered, instead of a family of decision problems, because the quality bound was part of the input to the decision problem. Moreover, the control parameter for these problems was chosen in such a way that it encoded the quality bound in some direct or indirect way, and this

choice was criticized in subsequent investigations [ST98]. Clearly, this criticism does not apply to the bounded satisfiability problems $k\text{SAT}(b)$.

For every integers $k \geq 3$, $n \geq 1$, and $b \geq 0$, let $\Pr_k[n, r, b]$ be the probability that a random formula φ in the space $F_k(n, r)$ is a “yes” instance of $k\text{SAT}(b)$, that is, there is a truth assignment that violates no more than b clauses of φ . We now have all the notation in place to formulate the following conjecture concerning phase transitions for the bounded satisfiability problems $k\text{SAT}(b)$.

Conjecture 4.2.1: For every integer $k \geq 3$ and every integer $b \geq 0$, there is a positive real number $r_{k,b}$ such that

- If $r < r_{k,b}$, then $\lim_{n \rightarrow \infty} \Pr_k[n, r, b] = 1$;
- If $r > r_{k,b}$, then $\lim_{n \rightarrow \infty} \Pr_k[n, r, b] = 0$.

Note that Chvátal and Reed’s conjecture [CR92a] concerning phase transitions for k -SAT, $k \geq 3$, is the special case of Conjecture 4.5.1 in which $b = 0$. Zhang [Zha01] raised the problem of finding the exact location of the phase transition for $3\text{-SAT}(b)$, which, in terms of the notation introduced here, amounts to first establishing Conjecture 4.5.1 for $k = 3$ and then determining the critical ratio $r_{3,b}$, for each b . Although far from solving these problems, the next result yields analytical upper bounds for the values of $r_{k,b}$; in particular, it demonstrates that for each $k \geq 3$, all critical ratios $r_{k,b}$ are bounded by a quantity that depends only on k .

4.3 Decision Versions of MAXSAT

There are several possible ways to define decision versions of MAXSAT. Some of these are outlined here for comparison with Zhang’s bounded SAT concept.

4.3.1 Goal: at least g clauses satisfied

Definition 4.3.1: $\text{MAX-}k\text{-SAT}(g, \varphi)$ Inputs: k -CNF formula φ , and g a goal which is zero or a positive integer. Output: Answer to the question: Is there an assignment that satisfies at least g clauses of φ ? ■

Definition 4.3.2: $g\text{-MAX-}k\text{-SAT}(\varphi)$ Input: k -CNF formula φ . Output: Answer to the question: Is there an assignment that satisfies at least g clauses of φ ? ■

This is the type of MAXSAT decision problem considered by Papadimitriou [Pap94]. He proves that the uniform version is NP-complete for $k = 2$, which is interesting since 2-SAT is P. He points out that for $k \geq 3$ the uniform version is trivially NP-complete since it is a generalization of 3-SAT. He does not mention the fixed version. It is obviously P, for all values of g .

4.3.2 Bound: at most b clauses violated

These are the “bounded” versions. Here the picture is similar the goal type but different.

Definition 4.3.3: $\text{BD-}k\text{-SAT}(b, \varphi)$ Inputs: k -CNF formula φ , and b a bound which is zero or a positive integer. Output: Answer to the question: Is there an assignment that does not violate more than b clauses of φ ? ■

Definition 4.3.4: $k\text{-SAT}(B)$ Input: k -CNF formula φ . Output: Answer to the question: Is there an assignment that does not violate more than b clauses of φ ? ■

With these versions the trivial versus interesting scale reverses. The uniform version $\text{BD-}k\text{-SAT}(b, \varphi)$ is clearly equivalent to $\text{MAX-}k\text{-SAT}(g, \varphi)$ with the 1 to 1 mapping: $g = |\varphi| - b$. So it is trivially NP-complete for all k and b .

Here the fixed version is the most interesting case, and it is the one considered by Zhang. It is NP-complete as shown in the following proposition.

4.4 NP-completeness of k -SAT(B)

Proposition 4.4.1: For every integer $k \geq 3$ and every integer $b \geq 0$, the decision problem k -SAT(B) is NP-complete.

Proof:

It is known that k -SAT is NP-complete for $k > 2$. We show that k -SAT(B) is NP and that k -SAT can be reduced to k -SAT(B).

It is straightforward to construct a polynomial time algorithm to check a putative witness for k -SAT(B), so clearly it is NP.

To show a reduction we will make use of a special gadget, a blanket function.

Definition 4.4.2: Blanket Function on k variables.

$\beta_k(x_1, x_2, \dots, x_k)$ the blanket function on k variables is the conjunction of the 2^k different k -disjuncts that can be formed from the k variables, x_1, x_2, \dots, x_k .

For example:

$$\begin{aligned}\beta_3(x_1, x_2, x_3) = & (x_1 \vee x_2 \vee x_3) \wedge \\ & (x_1 \vee x_2 \vee \neg x_3) \wedge \\ & (x_1 \vee \neg x_2 \vee x_3) \wedge \\ & (x_1 \vee \neg x_2 \vee \neg x_3) \wedge \\ & (\neg x_1 \vee x_2 \vee x_3) \wedge \\ & (\neg x_1 \vee x_2 \vee \neg x_3) \wedge \\ & (\neg x_1 \vee \neg x_2 \vee x_3) \wedge \\ & (\neg x_1 \vee \neg x_2 \vee \neg x_3).\end{aligned}$$

Clearly, β_k is not satisfiable, but more importantly for our purposes, for each possible assignment of the variables, *one and only one*, of the disjuncts in β_k is *unsatisfied*. ■

Now, given a k -SAT instance φ , we can build a new formula φ' by concatenating b β_k functions to it, each introducing k new variables. Let the new formula be:

$$\begin{aligned} \varphi' &= \varphi \wedge \\ &\quad \beta_k(x_{11}, x_{12}, \dots, x_{1k}) \wedge \\ &\quad \beta_k(x_{21}, x_{22}, \dots, x_{2k}) \wedge \\ &\quad \dots \\ &\quad \beta_k(x_{b1}, x_{b2}, \dots, x_{bk}). \end{aligned}$$

It will have at least b violated clauses for any assignment, so the answer to k -SAT(B) for φ' is yes if and only if φ is satisfiable. The b β_k functions can be concatenated in constant time so we have a polynomial time reduction of k -SAT to k -SAT(B) and for $k > 2$ these are NP-complete problems. ■

4.5 Upper and Lower Bounds for k -SAT(b)

We now have all the notation in place to formulate the following conjecture for the family of problems k -SAT(B), where $k \geq 3$ and $b \geq 0$.

Conjecture 4.5.1: For every integer $k \geq 3$ and every integer $b \geq 0$, there is a positive real number $r_{k,b}$ such that:

- If $r < r_{k,b}$, then $\lim_{n \rightarrow \infty} P(k\text{-SAT}(B)(\varphi) \text{ is YES}) = 1$.
- If $r > r_{k,b}$, then $\lim_{n \rightarrow \infty} P(k\text{-SAT}(B)(\varphi) \text{ is YES}) = 0$.

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

Any lower bound for the phase transition of k -SAT is a lower bound for k -SAT(B). The currently best known such bound is 3.42 [KKL02]. Furthermore, if a fixed critical ratio for the k -SAT transition exists it is a lower bound for k -SAT(B).

As for upper bounds, the first moment method yields the same bound for this problem as for k -SAT as seen in the following.

Proposition 4.5.2: *Let $\varphi_{k,n,r}$ be a randomly generated formula from the uniform selection of $m = rn$ clauses, with replacement, from all possible k -literal disjuncts of k distinct variables selected from n variables, then for all $k \geq 2$*

$$r > \frac{1}{k - \lg(2^k - 1)} \quad \text{implies} \quad \lim_{n \rightarrow \infty} \mathbf{P}\{\varphi_{k,n,r} \in k\text{-SAT}(b)\} \rightarrow 0$$

Proof: First we find the expected number of satisfying assignments for a randomly generated formula. There are a total of $2^k \binom{n}{k}$ possible clause types, so there are $(2^k \binom{n}{k})^m$ ways in which a formula can be made. For any particular assignment $\binom{n}{k}$ of the clause types will be false and $(2^k - 1) \binom{n}{k}$ will be true.

We are interested in formulas with a certain number, say i , of unsatisfied clauses. For any particular assignment, the number of ways in which a formula can have exactly i locations with unsatisfied clauses is $\binom{m}{i}$ so the number of ways in which an m clause formula can be made with exactly i unsatisfied clauses is:

$$\binom{m}{i} \binom{n}{k}^i \left((2^k - 1) \binom{n}{k} \right)^{m-i}$$

from which it follows that the number of ways in which formulas can be made that have b or less unsatisfying clauses is:

$$\sum_{i=0}^b \binom{m}{i} \binom{n}{k}^i \left((2^k - 1) \binom{n}{k} \right)^{m-i}$$

Now if we construct a table in which each row represents one of the possible formulas that can be generated, each column represents one of the possible assignments and a cell contains a 1 or a 0 depending on whether or not the formula for its row is satisfied by the assignment for its column, then

the above expression gives the total number of 1's that occur in each and every column of the table. So

the total number of 1's in the table is:

$$2^n \sum_{i=0}^b \binom{m}{i} \binom{n}{k}^i \left((2^k - 1) \binom{n}{k} \right)^{m-i}$$

And the number of 1's per formula is:

$$\frac{2^n \sum_{i=0}^b \binom{m}{i} \binom{n}{k}^i (2^k - 1)^{m-i} \binom{n}{k}^{m-i}}{2^{km} \binom{n}{k}^m}$$

This is the expected number of satisfying assignments for a randomly generated formula and we will call it \mathcal{E} , then

$$\begin{aligned} \mathcal{E} &= 2^n \sum_{i=0}^b \binom{m}{i} \frac{(2^k - 1)^m (2^k - 1)^{-i}}{2^{km}} \\ &= 2^n \sum_{i=0}^b \frac{\binom{m}{i}}{(2^k - 1)^i} \left(\frac{2^k - 1}{2^k} \right)^m \\ &= 2^n \left(\frac{2^k - 1}{2^k} \right)^m \sum_{i=0}^b \frac{\binom{m}{i}}{(2^k - 1)^i} \\ &= \left(2 \left(\frac{2^k - 1}{2^k} \right)^r \right)^n \sum_{i=0}^b \frac{\binom{m}{i}}{(2^k - 1)^i} \end{aligned}$$

and

$$\lim_{n \rightarrow \infty} \mathcal{E} \leq \lim_{n \rightarrow \infty} \frac{\mathcal{O}(n^b)}{\left(\frac{1}{2 \left(\frac{2^k - 1}{2^k} \right)^r} \right)^n}$$

and

$$r > \frac{1}{k - \lg(2^k - 1)} \text{ implies } \left(\frac{1}{2 \left(\frac{2^k - 1}{2^k} \right)^r} \right) > 1 \text{ implies } \lim_{n \rightarrow \infty} \mathcal{E} = 0$$

Now application of Markov's Inequality directly yields the result. ■

By combining the result of this proposition with the best known lower bound for the phase transition of k -SAT, we obtain the following bounds for the phase transitions of the bounded satisfiability problems 3-SAT(b).

Corollary 4.5.3: Let $b \geq 0$ be an integer.

- If $r < 3.42$, then $\lim_{n \rightarrow \infty} \mathbb{P}\{\varphi_{3,n,r} \in 3\text{-SAT}(b)\} \rightarrow 1$.
- If $r > \frac{1}{3 - \lg(7)} \approx 5.19$, then $\lim_{n \rightarrow \infty} \mathbb{P}\{\varphi_{3,n,r} \in 3\text{-SAT}(b)\} \rightarrow 0$.

Consequently, if $r_{3,b}$ exists, then $3.42 \leq r_{3,b} \leq 5.19$, regardless of the value of b .

4.6 2BDSAT(b) Phase Transition Same as 2SAT

Definition 4.6.1: BDSAT(b) is the decision problem that given, a CNF formula and a bound b which is zero or a positive integer, answers the question: is there an assignment that does not violate more than b clauses of the formula?

Definition 4.6.2: k BDSAT(b) is the decision problem that given, a k -CNF formula and a bound b which is zero or a positive integer, answers the question: is there an assignment that does not violate more than b clauses of the formula?

Proposition 4.6.3 Let φ_m be a random m -clause 2-CNF formula, uniformly selected from the standard distribution on n variables and let b be a positive integer. Then for any positive ϵ it must be the case that

$$\lim_{n \rightarrow \infty} \mathcal{P}\{\varphi_{(1+\epsilon)n} \text{ is BDSAT}(b)\} = 0.$$

We will prove this two different ways.

First proof.

Proof: Assume the premises of the proposition statement and further let

$$\delta = 1 + \epsilon - \frac{b}{n_0}$$

where n_0 is the smallest positive integer that will make b/n_0 less than ϵ .

Then for all $n \geq n_0$ we have the following:

$$\mathcal{P}\{\varphi_{(1+\epsilon)n} \text{ is BDSAT}(b)\} \leq \binom{(1+\epsilon)n}{b} \mathcal{P}\{\varphi_{(1+\epsilon)n-b} \text{ is SAT}\}$$

from the definition of bounded satisfiability and the fact that the probability of a disjunction can not exceed the sum of the probabilities of the disjuncts. Now

$$\binom{(1+\epsilon)n}{b} \mathcal{P}\{\varphi_{(1+\epsilon)n-b} \text{ is SAT}\} = \binom{(1+\epsilon)n}{b} \mathcal{P}\{\varphi_{(1+\epsilon-\frac{b}{n})n} \text{ is SAT}\}.$$

Note the clause to number of variables ratios considered are not fixed but change with each n , monotonically increasing from $1 + \delta$, for $n = n_0$, to arbitrarily close to $1 + \epsilon$ as n tends to infinity. But for any particular n , the SAT probability is monotonically decreasing with increasing ratios so

$$\binom{(1+\epsilon)n}{b} \mathcal{P}\{\varphi_{(1+\epsilon-\frac{b}{n})n} \text{ is SAT}\} \leq \binom{(1+\epsilon)n}{b} \mathcal{P}\{\varphi_{(1+\delta)n} \text{ is SAT}\}.$$

Now it is known [BBC⁺01] that for any positive δ independent of n

$$\mathcal{P}\{\varphi_{(1+\delta)n} \text{ is SAT}\} = \exp(-\Theta(\delta^3 n)).$$

So we conclude that

$$\mathcal{P}\{\varphi_{(1+\epsilon)n} \text{ is BDSAT}(b)\} \leq \frac{\Theta(n^b)}{\exp(\Theta(\delta^3 n))}$$

which clearly goes to 0 as n tends to infinity. ■

Second proof.

Proof: This proof makes use of three results established by Goerdt [Goe96c] in proving that 2SAT has a phase transition at 1.

Definition 4.6.4: A simple cycle is

Let X be a random variable for the number of simple cycles in a random formula graph and φ be a random formula with $r = 1 + \epsilon$ where $\epsilon > 0$. Then

1.

$$\lim_{n \rightarrow \infty} E(X) = 0$$

2.

$$\frac{V(X)}{E(X)^2} = o(1)$$

3. The average number of pairs of simple cycles that share common edges is asymptotically irrelevant with respect to the average number of pairs of cycles that do not have edges in common.

Now for bounded satisfiability we need to also consider some more complex structures.

Definition 4.6.5 A cycle complex of order k is a set of one or more simple cycles that form a maximally connected component that requires the removal of at least k shared edges to break up all of the simple cycles in the set.

Note, a simple cycle is a cycle complex of order 1, but a cycle complex of order 1 is not necessarily a simple cycle. For example a cycle complex of order 1 could be two simple cycles that share a single common edge.

Claim 4.6.6 A formula is 2BDSAT(b) satisfiable, iff, the sum of the orders of the cycle complexes in its corresponding formula graph is b or less.

Proof: The sum of the orders is the minimum number of edges whose removal would break up all cycles which corresponds to the minimum number of clauses whose removal would yield a satisfiable formula. ■

Claim 4.6.7: Let Y be a random variable for the sum of the orders of the cycle complexes in a random formula graph. Then

$$\lim_{n \rightarrow \infty} \mathcal{P}(Y = X) = 1$$

Proof: Follows from Goerdt's result 3 above. ■

Now as n tends to infinity:

$$\begin{aligned} \mathcal{P}\{\varphi \text{ is } 2\text{BDSAT}(b)\} &\leq \mathcal{P}\{Y < b + 1\} \\ &= \mathcal{P}\{X < b + 1\} \\ &\leq \mathcal{P}\{|X - E(X)| \geq E(X) - b\} \\ &= \mathcal{P}\{(X - E(X))^2 \geq (E(X) - b)^2\} \\ &\leq \frac{V(X)}{(E(X) - b)^2} \\ &= \frac{1}{(1 - \frac{b}{E(X)})^2} \frac{V(X)}{E(X)^2} \\ &= o(1) \end{aligned}$$

where the last step follows from Goerdt's results 1 and 2 above. ■

Corollary 4.6.8: $2\text{BDSAT}(b)$ for any positive integer b has a phase transition and its critical ratio is equal to 1.

Proof: Follows from the fact that 2SAT has a phase transition at 1 and that if a formula is satisfiable it is bounded satisfiable for any b . ■

4.7 Experiments

4.7.1 Method

We ran experiments for random 3CNF-formulas drawn from $n = 10, 15, 20, 25, 30$ and 35 variable spaces. We implemented a suitably modified version of the Davis-Logemann-Loveland-Putnam (DPLL) procedure and tested each random formula drawn to determine whether it is a “yes” instance of the bounded satisfiability problems 3-SAT(b), for $b = 3, 4$ and 5. For each (n, b) -pair, probability curves were determined by recording the average number of “yes” instances in samples of 1200 formulas, one sample for each possible ratio of clauses to variables up to a ratio of 15.

4.7.2 Algorithms

We modified the Davis-Putnam Procedure to make a BDSDP (Bounded SAT Davis-Putnam Procedure).

The BDSDP is a recursive function $\text{BDSDP}(\varphi, n, b)$ where φ is a Boolean CNF formula that contains no clause which is a tautology, n is the number of variables in the space considered, and b is the maximum number of clauses allowed to be unsatisfied. It is similar to the DPLL (Davis-Putnam-Logeman-Loveman) algorithm but modified to handle a bounded number of empty clauses.

- if φ contains more than b empty clauses, return 0, i.e. no
- if φ is empty or only contains empty clauses, return 1, i.e. yes
- if φ contains a unit clause $\{u\}$

return $\text{BDSDP}(\varphi|_{u \leftarrow 1}, n - 1)$

- otherwise choose any variable v in φ

return $\text{BDSDP}(\varphi|_{v \leftarrow 1}, n - 1) \vee \text{BDSDP}(\varphi|_{v \leftarrow 0}, n - 1)$

Where, since φ is a CNF, the restrictions $\varphi|_{x \leftarrow 1}$ and $\varphi|_{x \leftarrow 0}$ are as follows:

- $\varphi|_{x \leftarrow 1}$ contains all clauses in φ that do not contain x ; with the literal $\neg x$ removed if present
- $\varphi|_{x \leftarrow 0}$ contains all clauses in φ that do not contain $\neg x$; with the literal x removed if present

4.7.3 Results

The probability curves for 3-SAT(b) with $b = 3, 4$ and 5 are depicted in Figures 4.7, 4.9 and 4.11, respectively.

In these figures, we see that, as r increases, the probability curves start out in a region where the probability is 1, or close to 1, and then they transition to a region where the probability is 0, or close to 0. To make the terminology more precise, let us define the *location* of the transition for the probability of 3-SAT(b) to be the ratio r at which the probability is 0.5, and the *width* of the transition for the probability of 3-SAT(b) to be the difference in the r values between the ratios at which the probability falls from 0.9 to 0.1. As n increases, the probability curves move from right to left; moreover, both the location and the width of the transition appear to change with different values of n . For each fixed b , the width becomes smaller with increasing n , just as is the case for the probability curves for 3SAT in Figure 4.1. The location, however, moves dramatically to the left in sharp contrast with the behavior for 3SAT, where the location moves very slightly and converges to a value in the transition region where all the curves appear to intersect at one point. The magnitude of the leftward movement of the transition location with increasing n appears to become greater with increasing values of b . Thus, the phase transitions for the bounded satisfiability problems 3-SAT(b) emerge in a pattern that is novel and qualitatively different from that of the phase transitions for other NP-complete problems.

We also recorded the average cost of the modified DPLL-procedure for solving 3-SAT(b), where $b = 3, 4, 5$. Each of these problems is much harder on average than 3SAT; moreover, these problems are becoming harder on average as b increases. As a matter of fact, the experiments for $b = 5$

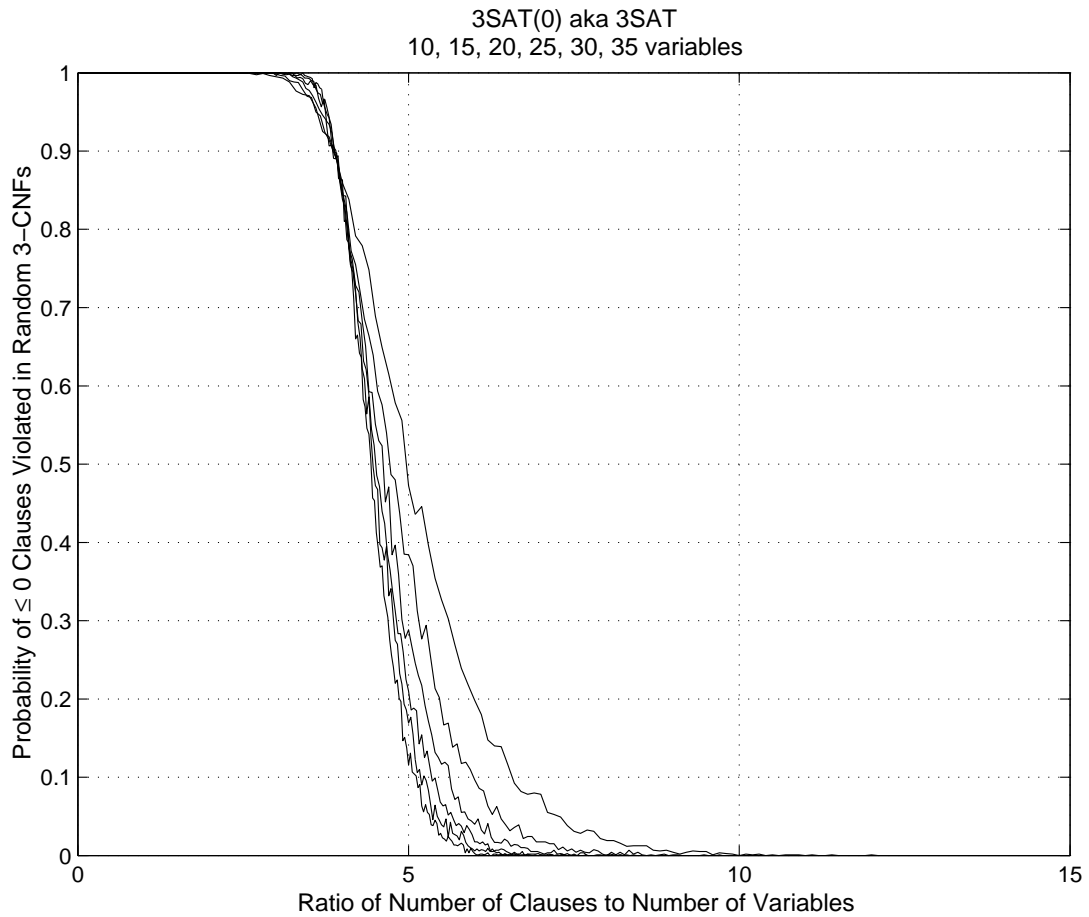


Figure 4.1: The probability curves for bounded satisfiability with $b = 0$. There is a curve for each of six different values of n . They show the well known 3SAT behavior. As n increases the transition regions of the curves become steeper and appear to have a common intersection point about which they rotate. The common hypothesis is that in the limit the curves approach a step function through this point and its ratio is the critical phase transition value for 3SAT, approximately 4.2. Note finite-size scaling with these small values for n estimates the ratio as approximately 4.00.

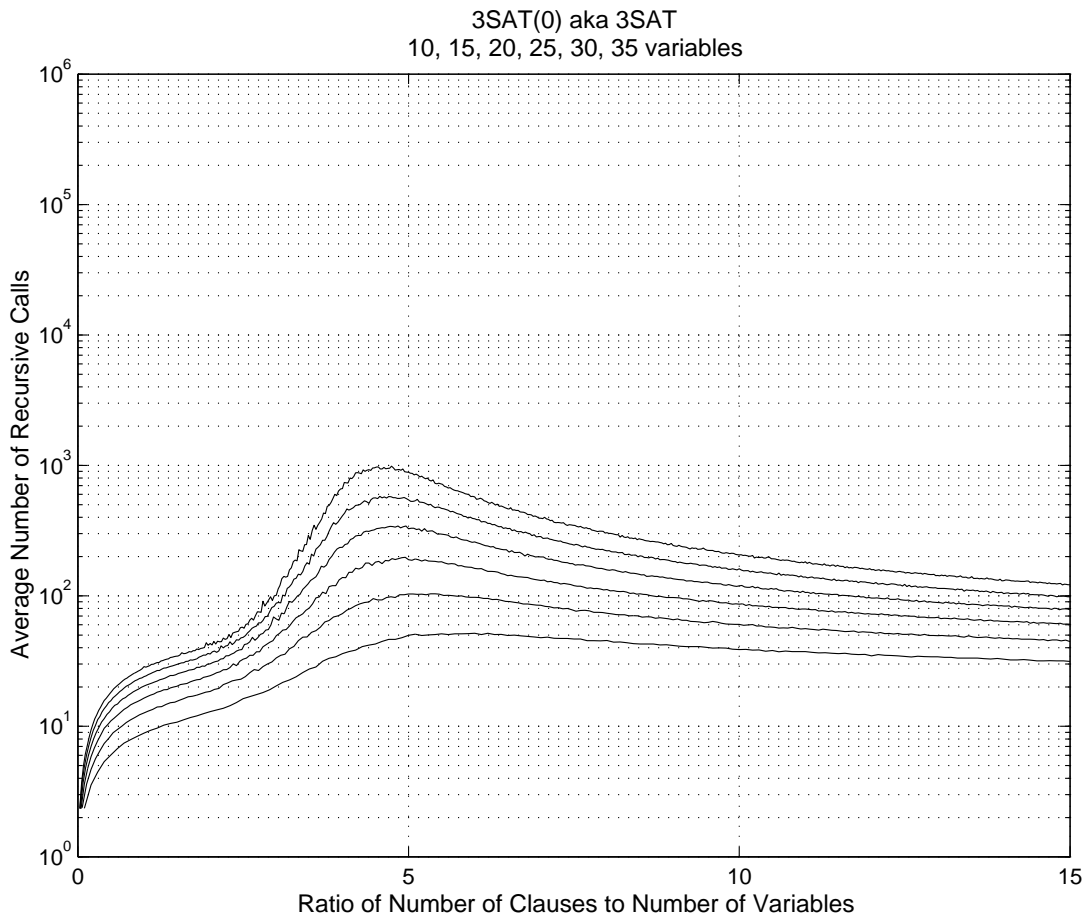


Figure 4.2: This set of curves plots the well known performance behavior of the DPLL procedure for 3SAT.

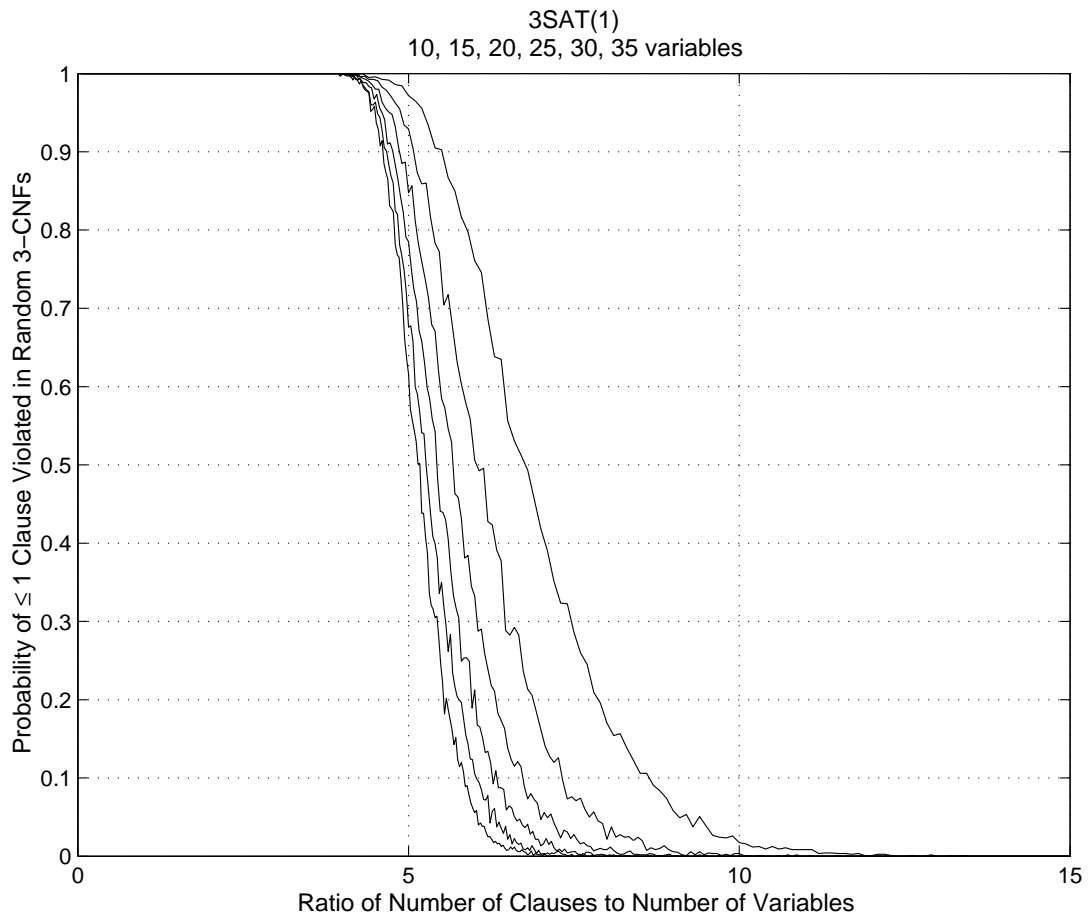


Figure 4.3: These are the probability curves for 3SAT(1). There is a curve for each of the same n values as in Figure 4.1. As n increases, the transition regions appear to steepen and distinctly move to the left. The steepening is similar to 3SAT behavior but here there is no apparent common rotation point. The transitions appear to be approaching a step function in the limit but it is not possible to visually determine the critical ratio. We used a form of finite-size scaling to estimate it as approximately 4.11

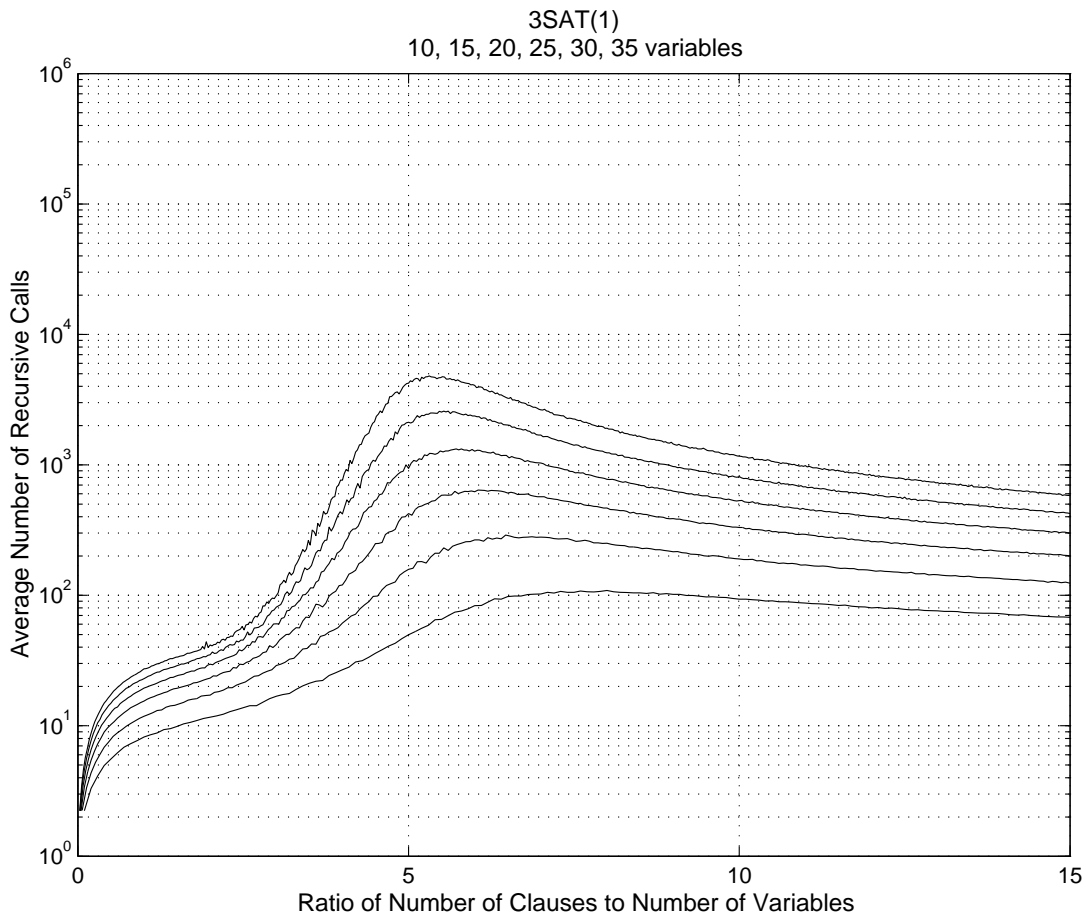


Figure 4.4: This set of curves plots the algorithmic costs for 3SAT(1). The behavior is similar to 3SAT in that cost increases with increasing n ; moreover, cost exhibits the so-called “easy/hard/easy” pattern as the ratio increases. As would be expected the costs for bounded satisfiability are much higher than for satisfiability. Note that the movement of the peak locations, as n increases, is quite marked and corresponds to the leftward movement of the transition regions in the probability curves.

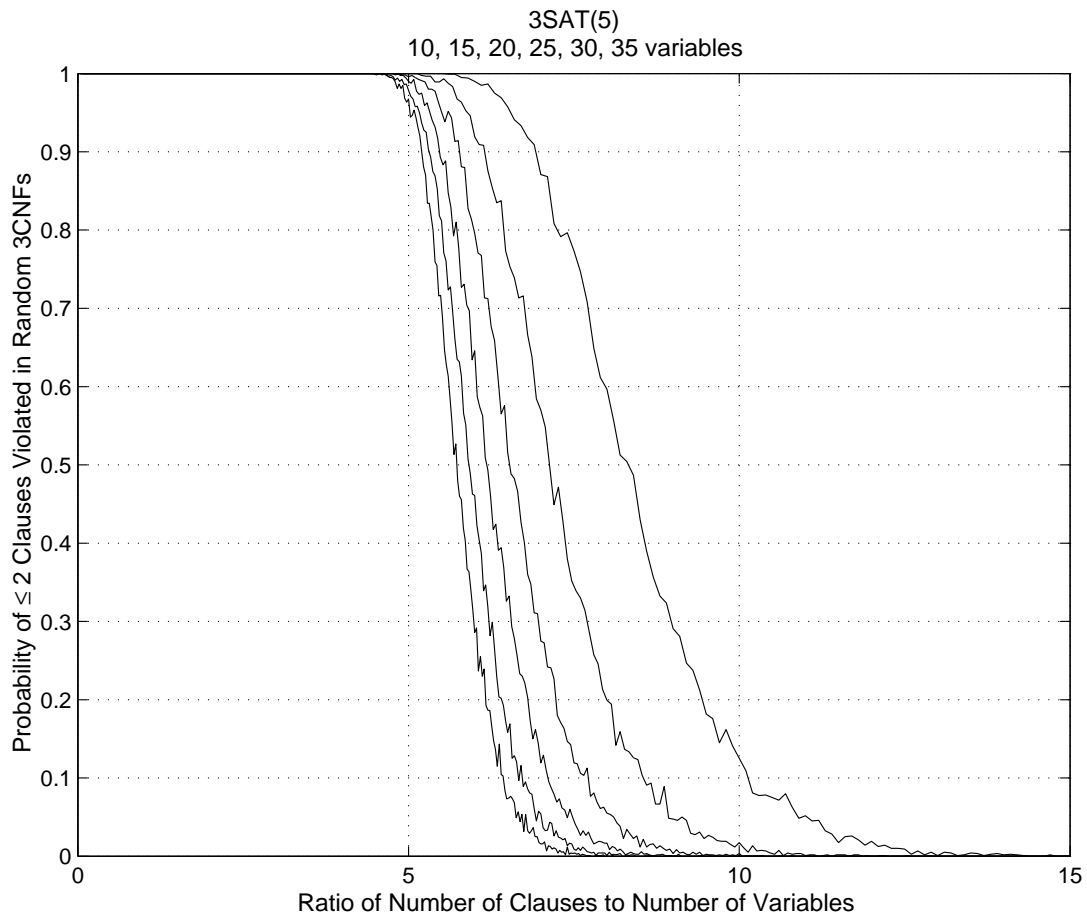


Figure 4.5: These are the probability curves for 3SAT(2). There is a curve for each of the same n values as in Figure 4.1. As n increases, the transition regions appear to steepen and distinctly move to the left. The steepening is similar to 3SAT behavior but here there is no apparent common rotation point. The transitions appear to be approaching a step function in the limit but it is not possible to visually determine the critical ratio. We used a form of finite-size scaling to estimate it as approximately 4.20

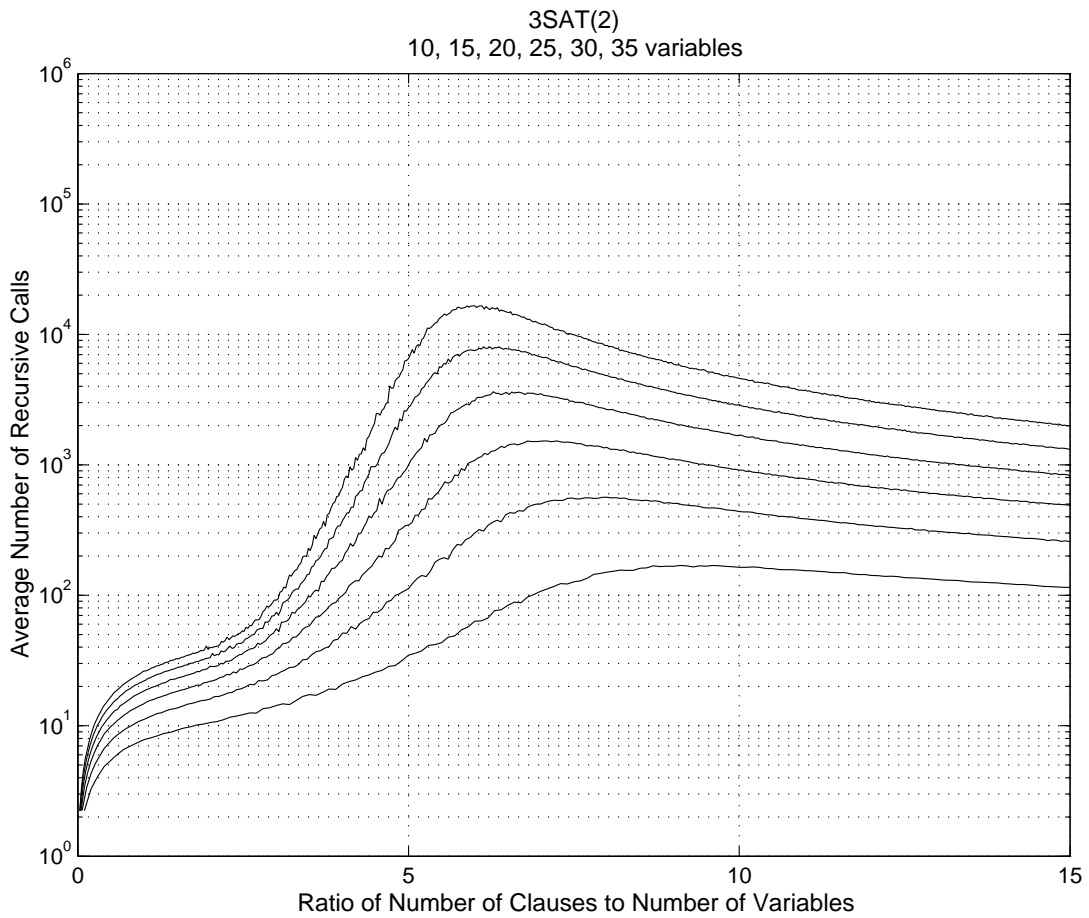


Figure 4.6: This set of curves plots the algorithmic costs for 3SAT(2). The behavior is similar to 3SAT in that cost increases with increasing n ; moreover, cost exhibits the so-called “easy/hard/easy” pattern as the ratio increases. As would be expected the costs for bounded satisfiability are much higher than for satisfiability. Note that the movement of the peak locations, as n increases, is quite marked and corresponds to the leftward movement of the transition regions in the probability curves.

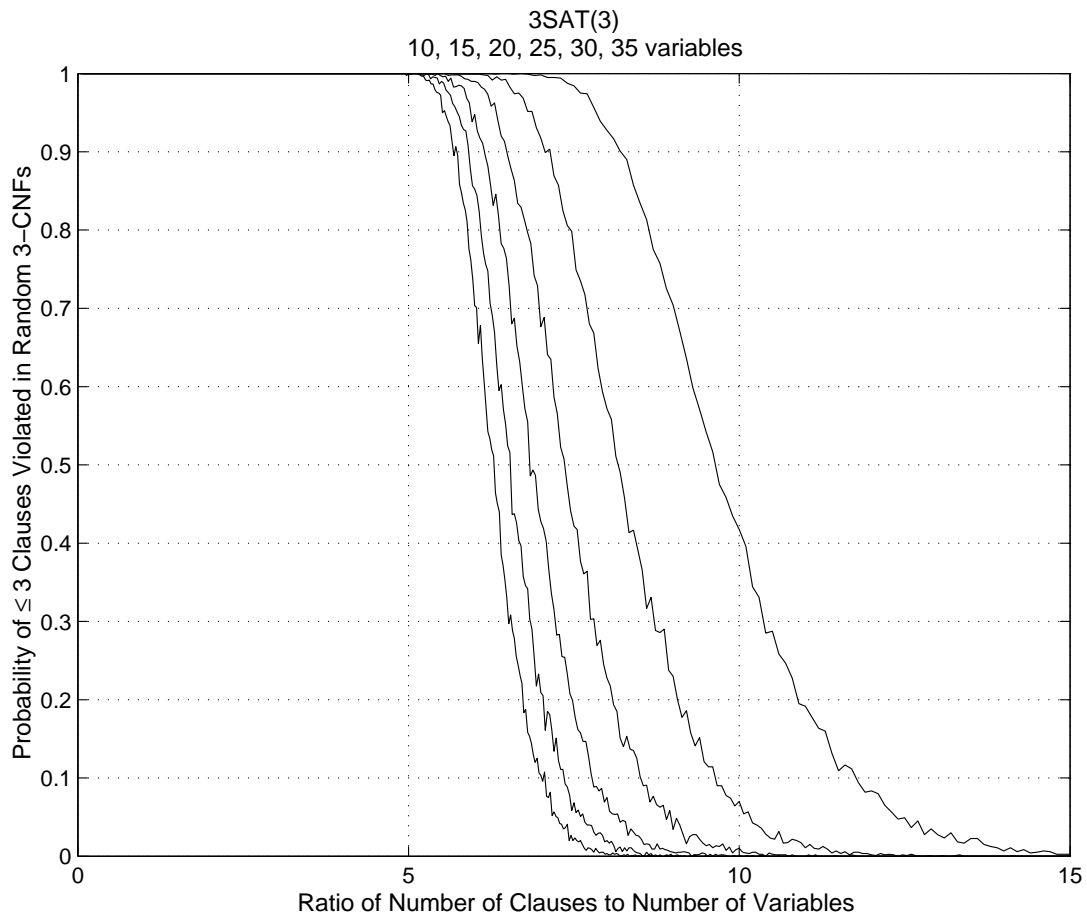


Figure 4.7: These are the probability curves for 3SAT(3). There is a curve for each of the same n values as in Figure 4.1. As n increases, the transition regions appear to steepen and distinctly move to the left. The steepening is similar to 3SAT behavior but here there is no apparent common rotation point. The transitions appear to be approaching a step function in the limit but it is not possible to visually determine the critical ratio. We used a form of finite-size scaling to estimate it as approximately 4.29

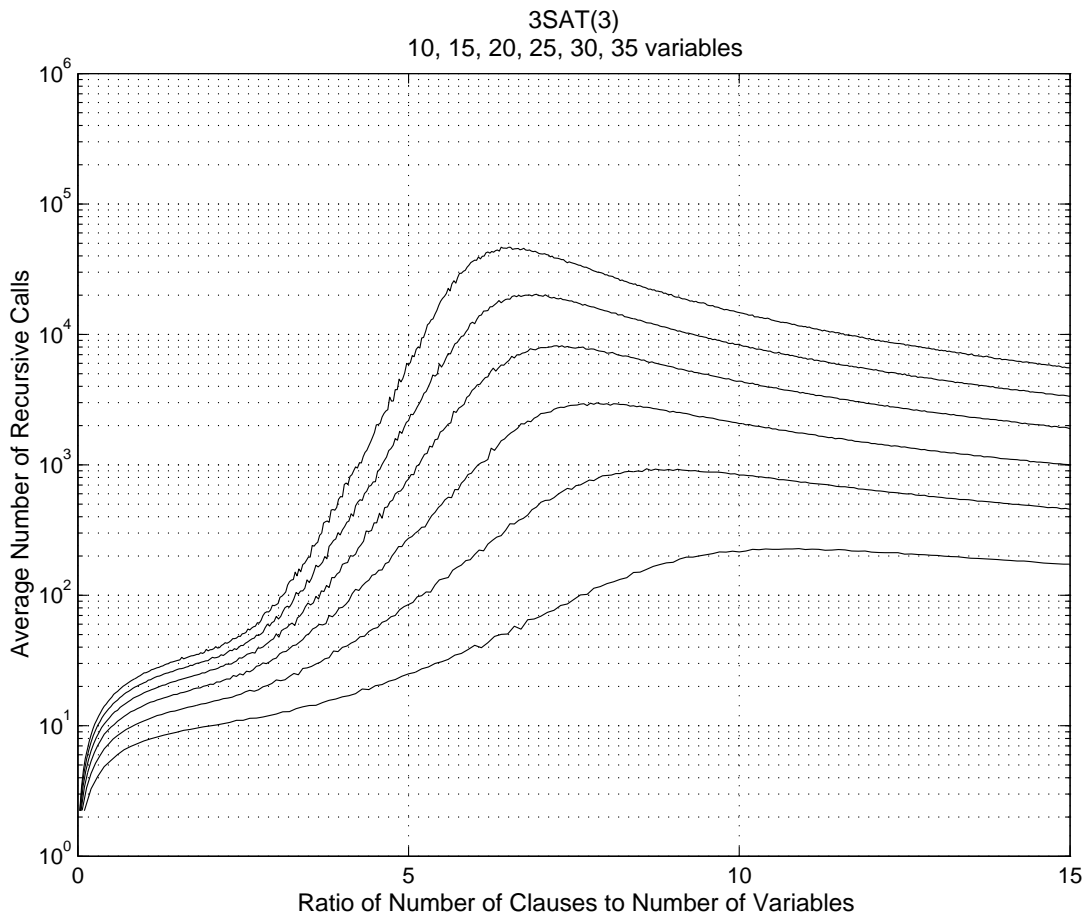


Figure 4.8: This set of curves plots the algorithmic costs for 3SAT(3). The behavior is similar to 3SAT in that cost increases with increasing n ; moreover, cost exhibits the so-called “easy/hard/easy” pattern as the ratio increases. As would be expected the costs for bounded satisfiability are much higher than for satisfiability. Note that the movement of the peak locations, as n increases, is quite marked and corresponds to the leftward movement of the transition regions in the probability curves.

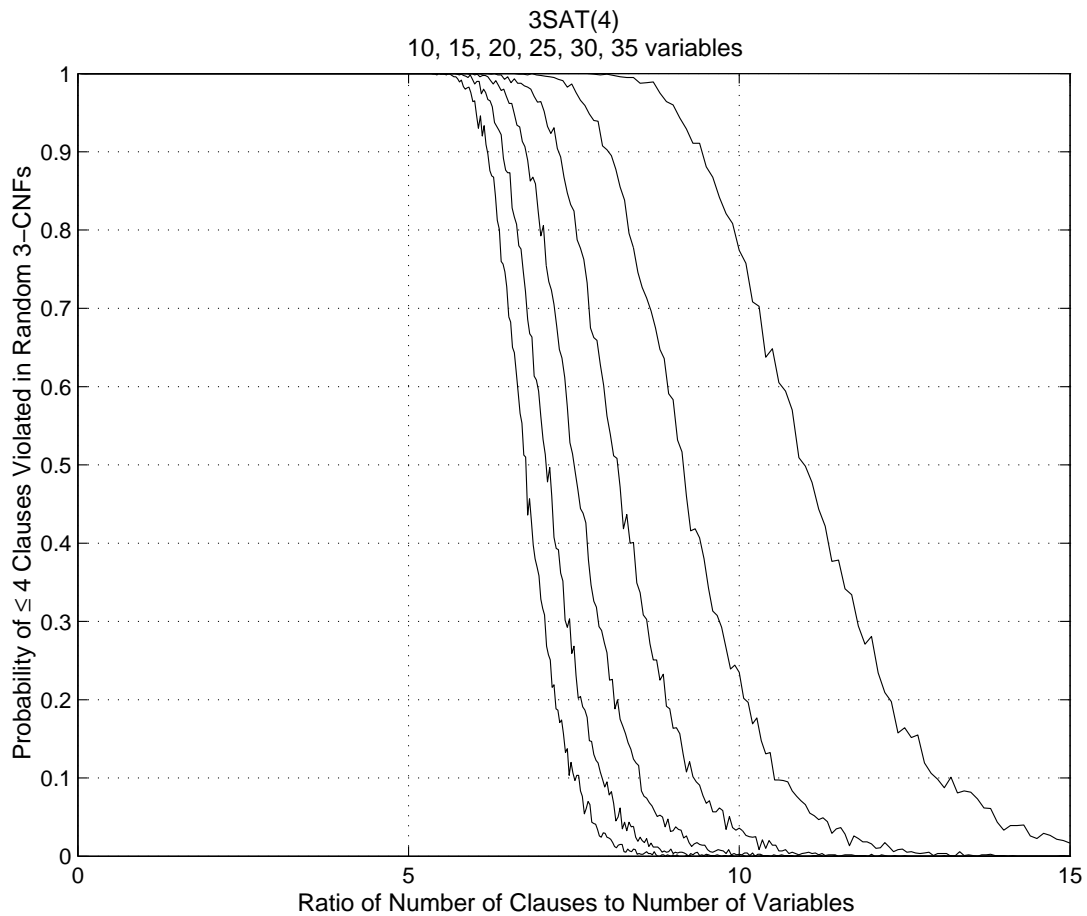


Figure 4.9: These are the probability curves for 3SAT(4). There is a curve for each of the same n values as in the previous figures. Again, as n increases, the transition regions appear to steepen and distinctly move to the left. The steepening is similar to 3SAT behavior but here there is no apparent common rotation point. The transitions appear to be approaching a step function in the limit but it is not possible to visually determine the critical ratio. We used a form of finite-size scaling to estimate it as approximately 4.32

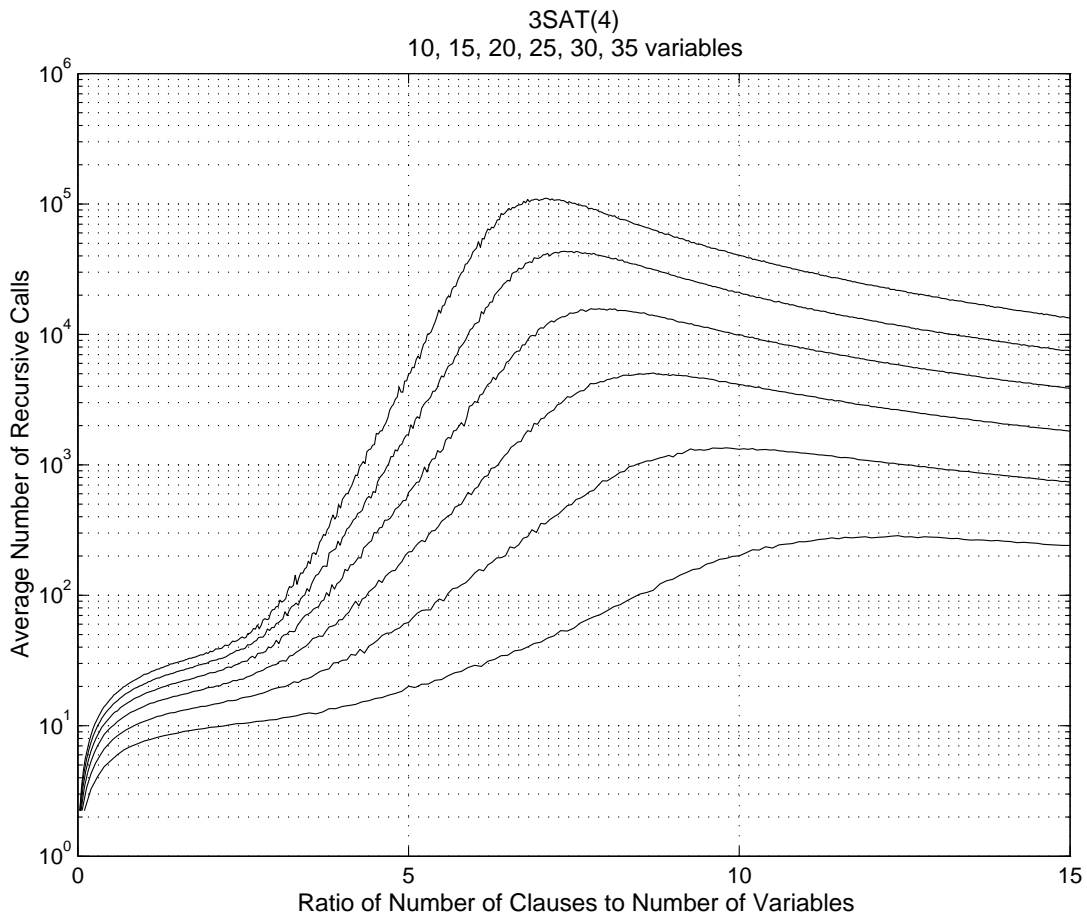


Figure 4.10: This set of curves plots the algorithmic costs for 3SAT(4). The behavior is similar to 3SAT in that cost increases with increasing n ; moreover, cost exhibits the so-called “easy/hard/easy” pattern as the ratio increases. As would be expected the costs for bounded satisfiability are much higher than for satisfiability. Note that the movement of the peak locations, as n increases, is quite marked and corresponds to the leftward movement of the transition regions in the probability curves.

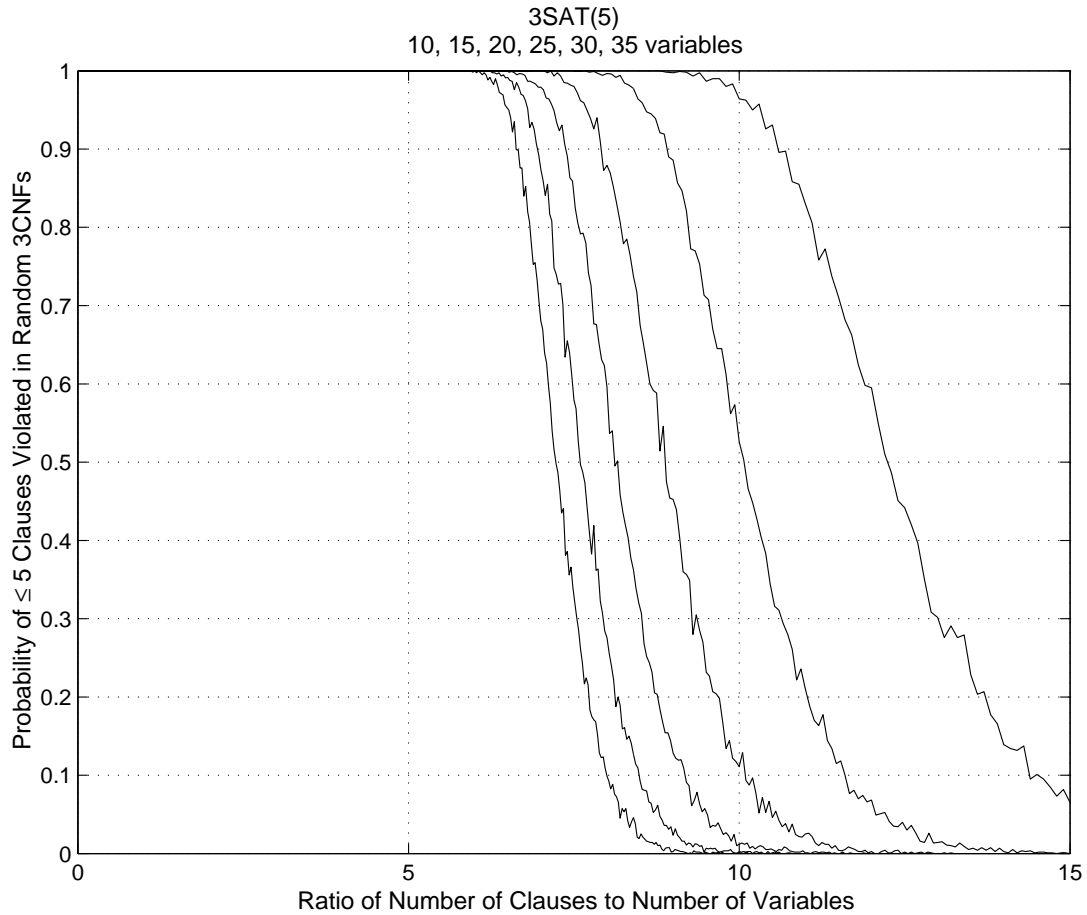


Figure 4.11: These are the probability curves for 3SAT(5). There is a curve for each of the same n values as in the previous figures. Again, as n increases, the transition regions appear to steepen and distinctly move to the left. The steepening is similar to 3SAT behavior but here there is no apparent common rotation point. The transitions appear to be approaching a step function in the limit but it is not possible to visually determine the critical ratio. We used a form of finite-size scaling to estimate it as approximately 4.43. Note the analytical upper bound is approximately 5.19.

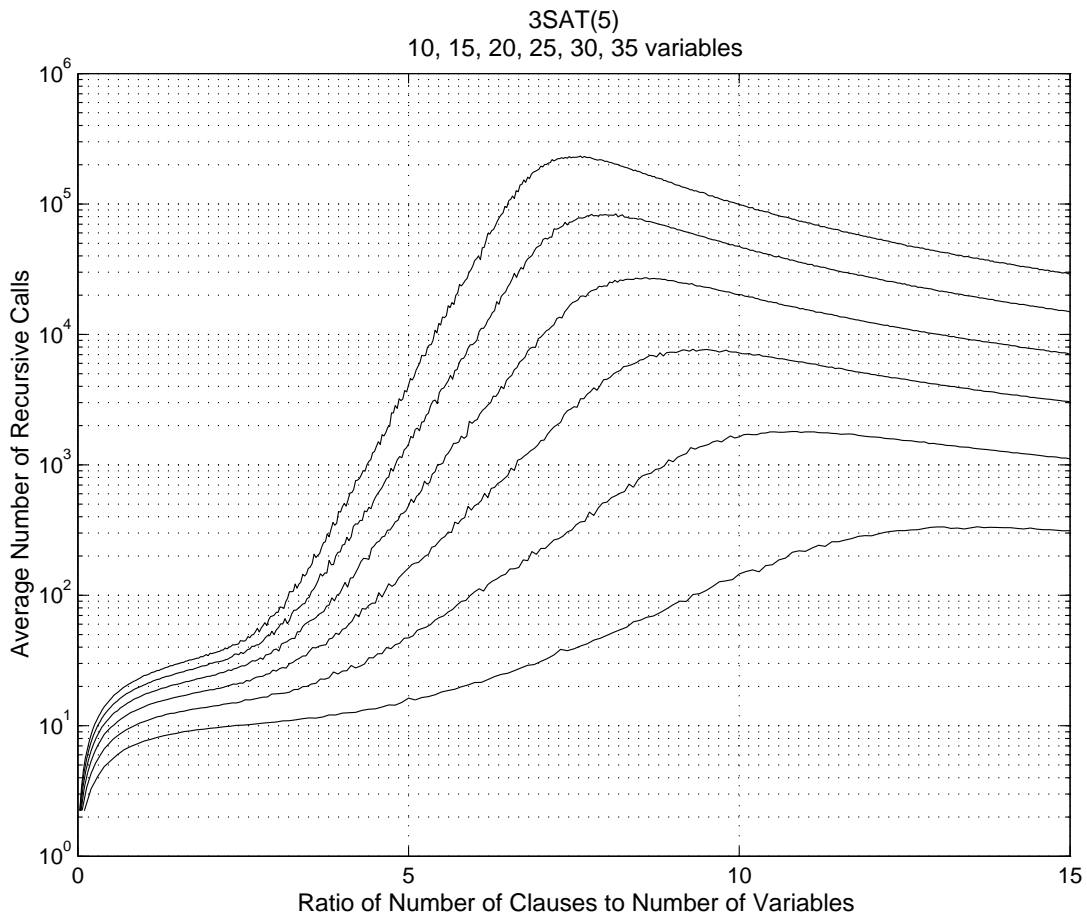


Figure 4.12: This set of curves plots the algorithmic costs for 3SAT(5). The behavior is similar to 3SAT in that cost increases with increasing n ; moreover, cost exhibits the so-called “easy/hard/easy” pattern as the ratio increases. As would be expected the costs for bounded satisfiability are much higher than for satisfiability. Note that the movement of the peak locations, as n increases, is quite marked and corresponds to the leftward movement of the transition regions in the probability curves.

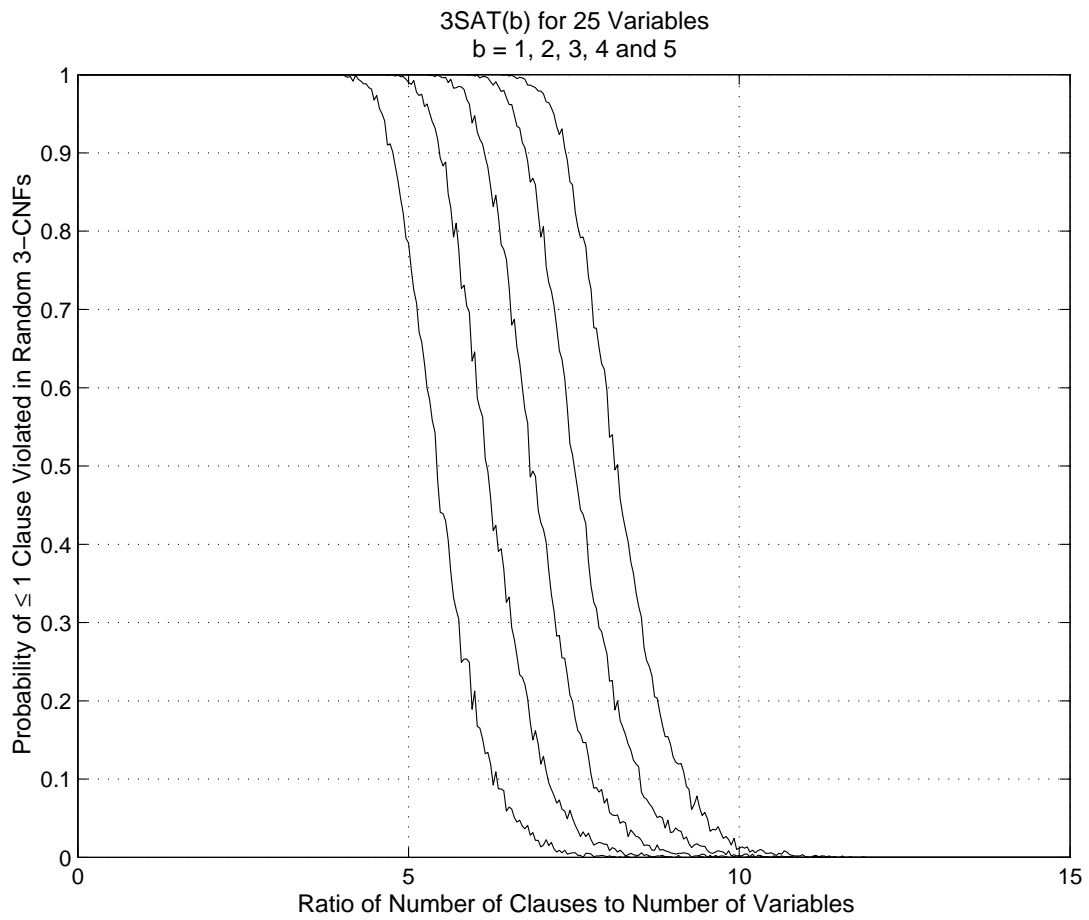


Figure 4.13: This figure shows the apparently separate “phase transitions” for different values of b when looking at probability curves for a fixed n , in this case 25 variables. The transition region moves from left to right with increasing b . This type of behavior was previously reported for $b = 5, 10, 15$, and 20 [Zhang, 2001].

and for $n = 35$ took several weeks to complete. Figure 4.2 depicts the average cost of solving 3SAT, while Figure 4.12 depicts the average cost of solving 3-SAT(5).

In Figure 4.12, the performance curves for 3-SAT(5) move from lower to higher with increasing n and, as has been observed with 3SAT, the peaks in cost appear to correspond to the location of the transitions. The peaks move from right to left in synchrony with the movement of the location of the transition for 3-SAT(b); this movement is more dramatic than in the case of 3SAT.

4.7.4 Finite-size Scaling

The locations of the transition in the probability curves for 3-SAT(b) appear to approach a limiting critical ratio $r_{3,b}$ that is within the analytically derived upper and lower bounds in Corollary 4.5.3. Nonetheless, it is not clear what the critical ratio $r_{3,b}$ actually is, nor can it be estimated by visual inspection since no two probability curves for 3-SAT(b) intersect. A finite-size scaling analysis of the data, assuming a power law of the form $\frac{(r-r_{3,b})n^{\nu_{3,b}}}{r_{3,b}}$, allowed us to obtain estimates for both the critical ratio $r_{3,b}$ and for the exponent $\nu_{3,b}$, where $b = 3, 4$ and 5 .

Kirkpatrick and Selman [KS94b], estimated the value of the critical ratio r_k by visually estimating the point at which the probability curves for k -SAT appear to intersect, and then applying regression techniques to determine the best exponent, ν_k , in the power law for k -SAT. This cannot be done with the bounded satisfiability problems 3-SAT(b), since the probability curves do not intersect, but appear to be moving to the left as the number n of variables increases. Nevertheless the finite-size scaling hypothesis can still be tested with a more elaborate procedure. Our goal was to test whether it is possible to find values for $r_{3,b}$ and $\nu_{3,b}$ such that when the ratio r is rescaled with the above power law, the probability curves for the different values n of the number of variables collapse to a single curve. Using routines for interpolation and regression from Matlab, we stepped through small increments of putative $r_{3,b}$ values and measured how well the curves collapsed to a single curve by calculating the sum of their pairwise squared differences when accordingly transformed. The estimated $r_{3,b}$ was taken

Table 4.1: Finite-size Scaling Results

b	$r_{3,b}$	$\nu_{3,b}$
0	4.00	0.63
1	4.11	0.70
2	4.20	0.74
3	4.29	0.76
4	4.32	0.80
5	4.43	0.84

to be the value that minimized this difference measure. We found this technique to be well-behaved, in that it exhibited a clear minimum and gave much more precision than visually judging how well the curves appeared to collapse to a single curve.

The results are shown in figures 4.14, 4.15, 4.16, 4.17, 4.18, and 4.19.

To further validate this technique, we checked it on the phase transition of 2SAT, which has been analytically determined to occur at the critical ratio $r_2 = 1$ [CR92a, Fer92a, Goe96a]. The agreement was very good, namely, the estimate for r_2 was 0.98.

The results of the finite-size scaling analysis for 3-SAT(b), where $b = 0, 1, 2, 3, 4$ and 5 , are shown in Table 4.1. Note that these results are consistent with the analytical upper and lower bounds for $r_{3,b}$ in Corollary 4.5.3, that the $r_{b,k}$ are in the region between r_k for k -SAT and the Markov upper bound for k -SAT. This consistency is the most that can be reasonably concluded from the finite-size scaling analysis. The technique provides a more principled way to make the estimates than eye-balling, but no great accuracy should be attributed to the estimates in spite of their apparent precision. Although the finite-size scaling results suggest that the locations of the phase transitions of 3-SAT(b) are different for different b 's, it may still be the case that they all coincide with the location for 3SAT.

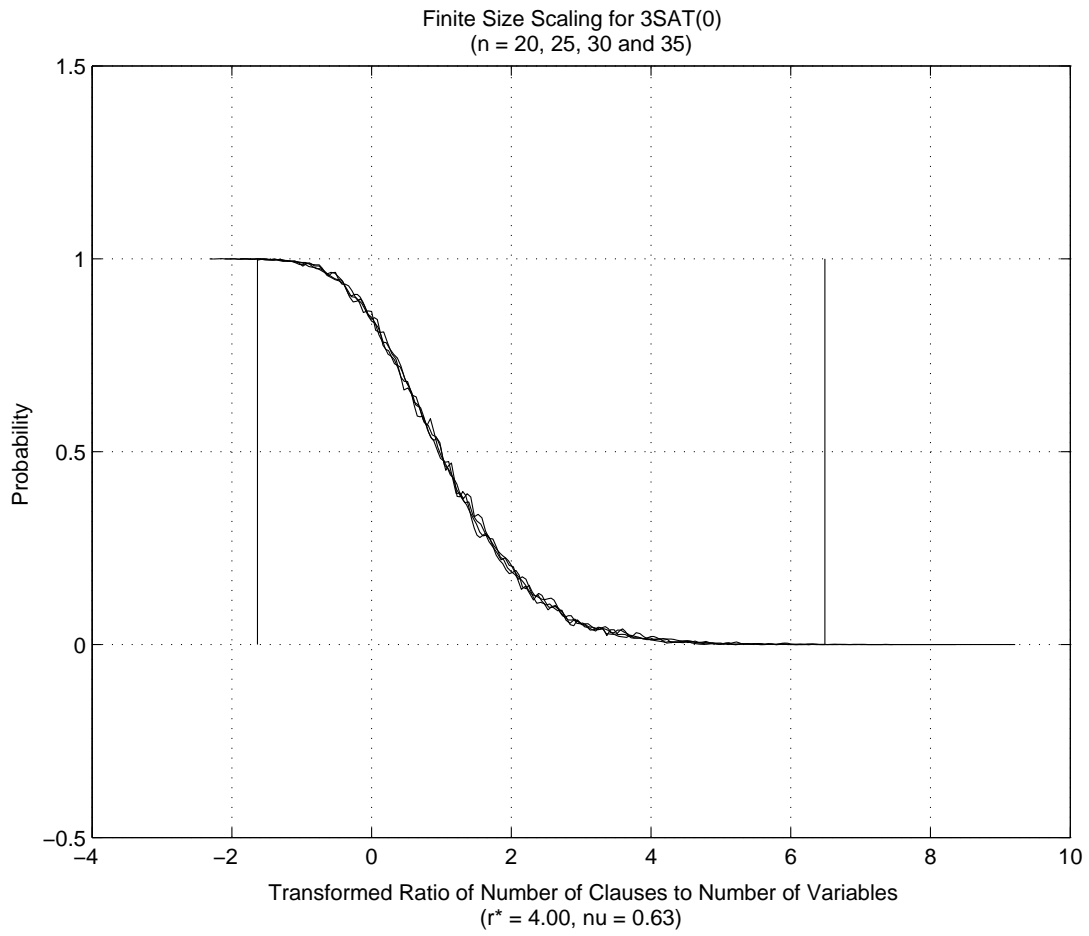


Figure 4.14:

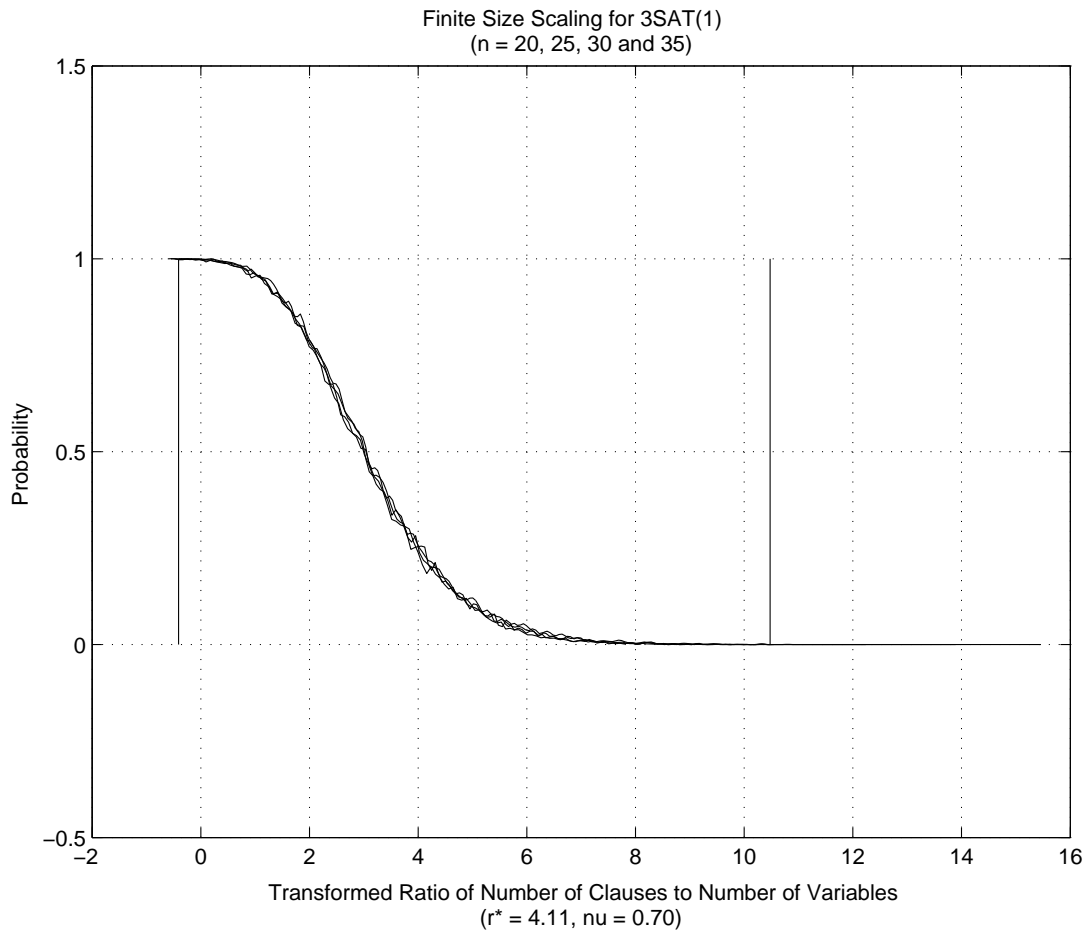


Figure 4.15:

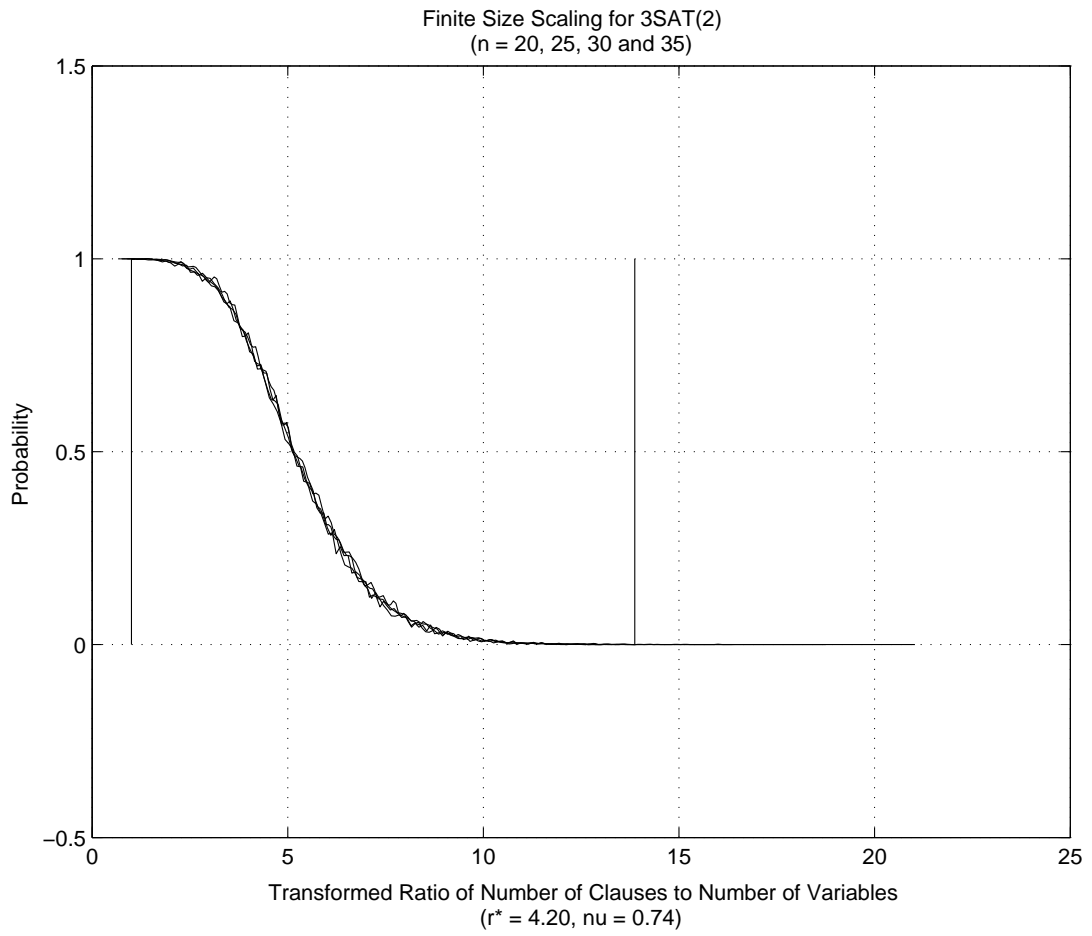


Figure 4.16:

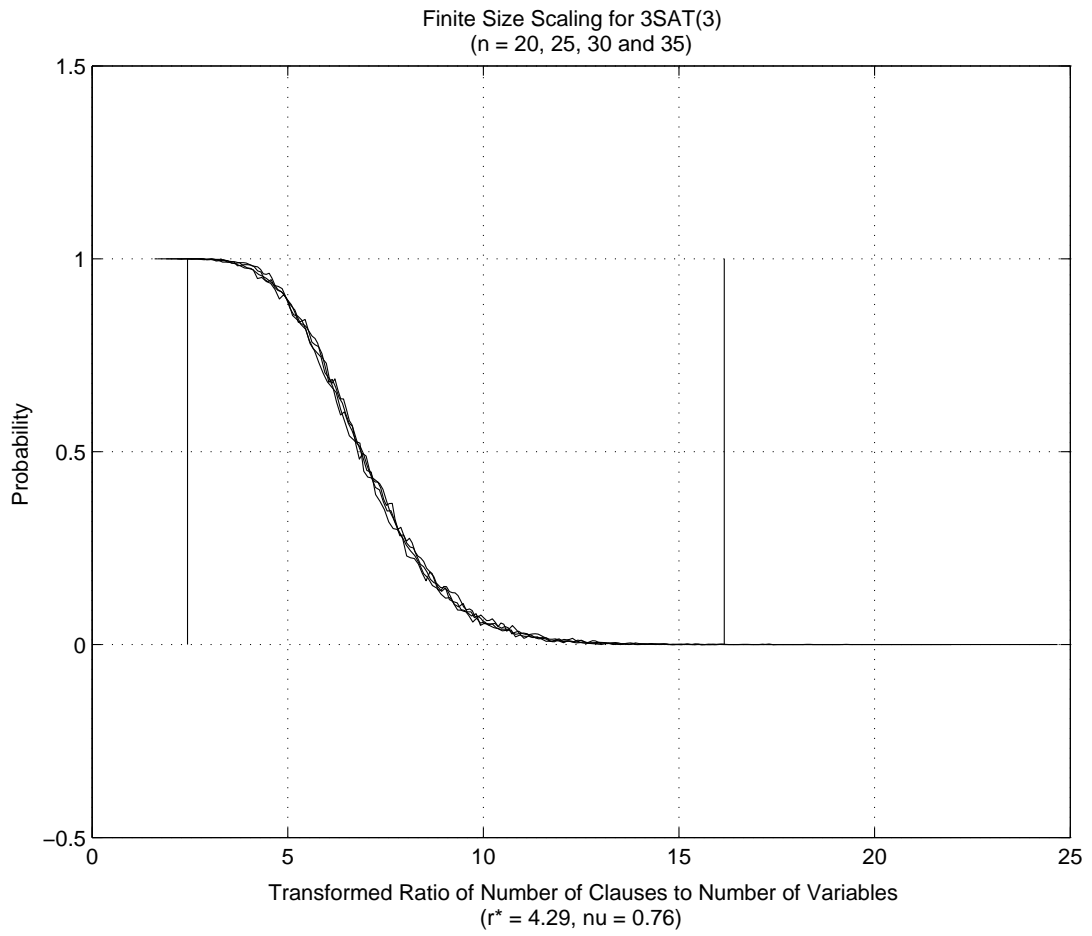


Figure 4.17:

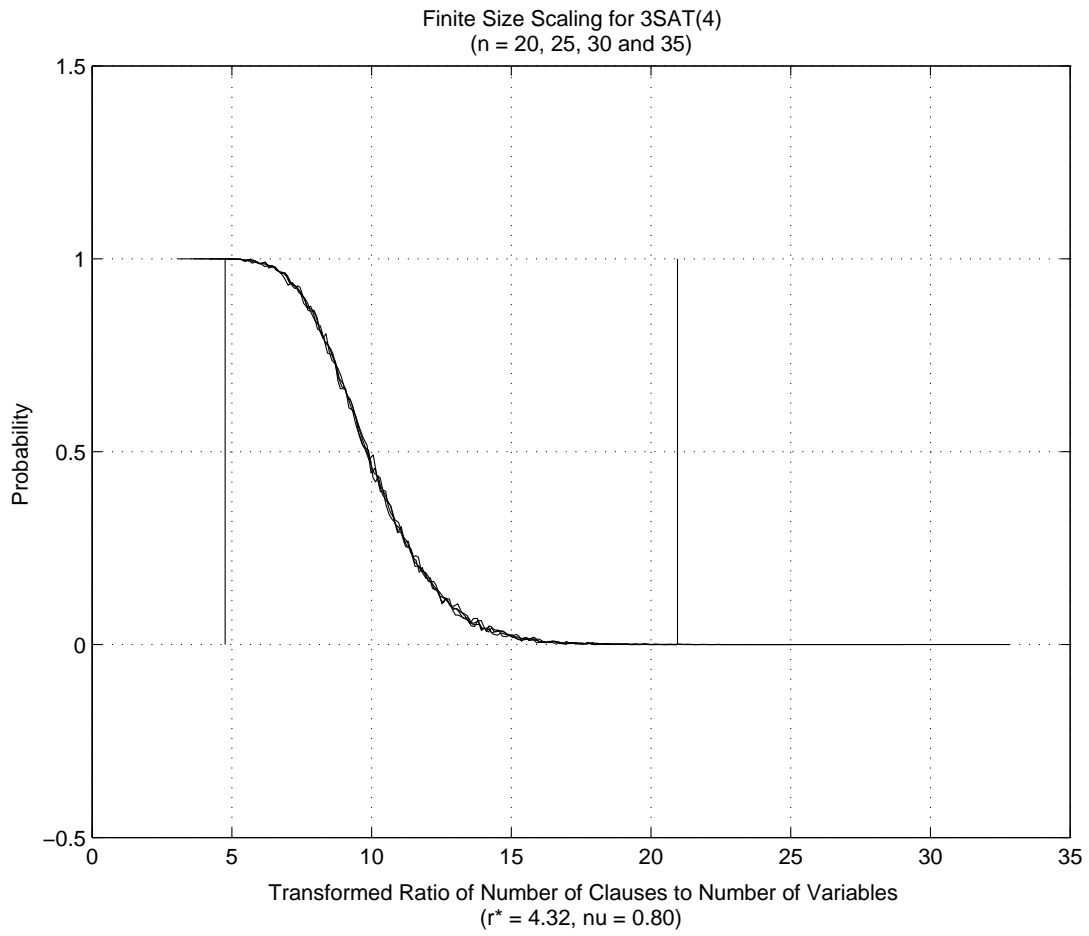


Figure 4.18:

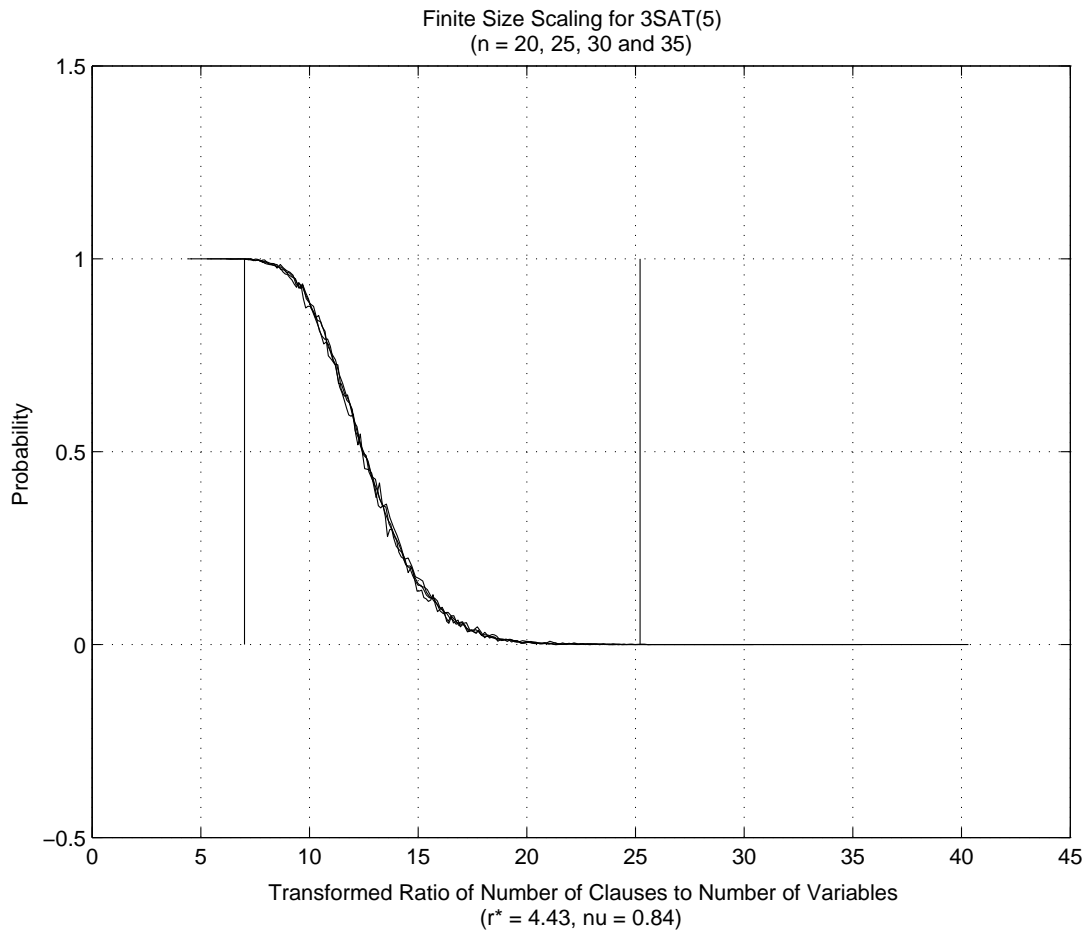


Figure 4.19:

In conclusion of this part, the results reported here advance the understanding of the phase transitions for the family of bounded satisfiability problems $3\text{-SAT}(b)$, introduced by Zhang [Zha01].

Our main analytical finding is that the phase transitions of all $3\text{-SAT}(b)$ problems must occur within a narrow region, regardless of how large the value of b is. Furthermore, we showed that the analytical phase transition location for 2-SAT is also an exact phase transition for $2\text{-SAT}(b)$. Moreover, our experiments revealed that the phase transitions for these problems occur in a remarkable way. Specifically, unlike 3SAT , the probability curves for $3\text{-SAT}(b)$ do not have a quasi-common intersection point about which they rotate as they become steeper with increasing n . Instead, they move rapidly to the left toward the narrow region that the analysis predicts.

Identifying the exact locations of the phase transitions for $3\text{-SAT}(b)$ remains an open problem, which is at least as hard as identifying the location of the phase transition for 3SAT . Equally hard, appears to be the problem of analytically establishing whether these locations are separated or coincide. Nonetheless, it may be possible to analytically obtain tighter bounds for these locations using some of the sophisticated techniques that have been quite successful in finding tighter bounds for the phase transition of 3SAT .

Part V

Generalized Satisfiability Problems

Chapter 5

Generalized Satisfiability Problems

5.1 Introduction

Several variants of satisfiability have been investigated with respect to complexity [GJ79]. Variants are defined by restricting the type of literals that can be in the clauses or interpreting the meaning of clauses differently from disjunction, or both. Some examples are: MONOTONE SAT where all the literals in each clause must be either all positive or all negative; 1-IN- k -SAT where the meaning is that one and only one of the literals is true; POSITIVE-1-IN- k -SAT where it is the same as 1-IN- k -SAT but all the literals must be positive; and NAE-SAT where the meaning is not all of the literals have the same truth value. Phase transitions have also been observed and analyzed in variants of satisfiability. Recently, Achlioptas et al [ACIM01] proved that 1-IN- k -SAT has a sharp threshold and with an exact location at $1/\binom{k}{2}$, and analyzed NAE 3-SAT proving it has a sharp threshold and lower and upper bounds of 1.514 and 2.215.

The number of possible variants of satisfiability, although bounded, is certainly very large, considering there are 2^{2^k} possible Boolean functions that can be defined on $\{0, 1\}^k$. Schaefer, in 1978, developed a framework, called generalized satisfiability, that includes all possible variants and with

respect to complexity proved a remarkable dichotomy theorem. It asserts that, assuming $P \neq NP$, the generalized satisfiability problems are, exclusively, either NP-complete or decidable in polynomial time. Moreover he identified five classes of clauses that can be used to definitively determine on which side of the dichotomy a given problem lies. If a variant problem allows only clauses exclusively of one of the Schaefer classes, then it is decidable in P time; otherwise it is NP -complete.

Given Schaefer's general framework for variants of satisfiability and its success in addressing questions of complexity, it is natural to explore phase transition phenomena of random satisfiability in his framework as well. In 2000, Istrate [Ist00] reported results on producing a classification of the order of phase transitions in generalized satisfiability problems. In 2003, Creignou and Daudé [CD03] reported results on classifying phase transitions in generalized satisfiability problems including a uniform approach to establishing lower and upper bounds for the location of the transition in these problems.

In the following we develop a uniform approach to establishing upper bounds for the generalized satisfaction problems in Schaefer's framework that produces bounds tighter than those that can be obtained from the previously reported uniform approach [CD03]. First, before going to the general framework, we do a first moment analysis of the upper bound for monotone k -SAT to show an example of the kind of complications that arise in the satisfiability variants, then we generalize the method to handle any type of generalized SAT problem. Finally, using the generalized technique we illustrate its use, finding bounds for 1-IN- k -SAT and POSITIVE-1-IN- k -SAT. Results of experiments and finite-size scaling as evidence of how the phase transition emerges and how it relates to the predicted bounds, is reported for each problem immediately after its upper bound analysis. The analytical bound for MONOTONE-3-SAT turns out to be the same as for 3-SAT, which suggests that the phase transition may be the same. The experiments bear this out showing a striking resemblance to 3-SAT for the MONOTONE-3-SAT problem.

5.2 An Upper Bound for Phase Transitions in Random Monotone k -CNF Formulas

Definition 5.2.1: The MONOTONE- k -SAT problem for k -CNF formulas is a decision problem which has as input a k -CNF formula in which all clauses are monotone, i.e., all the literals are either all positive or all negative, and which has as output YES if there exists at least one assignment of values to the variables in the formula, such that all of the clauses in the formula are satisfied, and the answer is NO, if for all possible assignments, there are always one or more clauses which are not satisfied. ■

Closely related more general decision problems, #MONOTONE- k -SAT($\geq 2^{\alpha n}$) and #THRESHOLD- k -SAT($\geq 2^{\alpha n}$), where $0 \leq \alpha < 1$ is a real number, ask the question: does there exist a number of satisfying assignments that is greater than or equal to the counting threshold value $2^{\alpha n}$? We will find the first moment upper bound for the phase transitions of #MONOTONE- k -SAT($\geq 2^{\alpha n}$), then the result for MONOTONE- k -SAT is an immediate corollary. We will use the following notation:

σ	A truth assignment.
$\Sigma_n = \{\sigma_1, \sigma_2, \dots, \sigma_{2^n}\}$	The set of all possible truth assignments on n variables.
φ	A Boolean formula.
$\mathcal{N}_\sigma(\varphi)$	The number of satisfying assignments of φ .
$\mathcal{N}_\varphi(\sigma)$	The number of formulas satisfied by assignment σ .
$\mathcal{F}_{n,m}^k$	The family of m -clause monotone CNF formulas which can be formed by concatenating m randomly selected clauses without replacement.
$P(A)$	The probability of event A .
$E(V)$	The expectation of random variable V .

Note that in a space of n variables the total number of monotone k -disjunct types is $2^{\binom{n}{k}}$ and the number of distinct monotone formulas with m clauses, $|\mathcal{F}_{n,m}^k|$, is $(2^{\binom{n}{k}})^m$. The number of the disjunct types

made false by any particular assignment to the variables depends on the number of variables assigned true. Since it takes k variables that are all true or all false to be able to make a disjunct false, the number of disjunct types made false by an assignment is $\binom{i}{k} + \binom{n-i}{k}$, where i is the number of variables made true by the assignment.

Upper Bound

Theorem 5.2.2: *If a critical r^* exists for #MONOTONE- k -SAT($\geq 2^{\alpha n}$), where $0 \leq \alpha < 1$ is a real, it must be less than $\frac{1-\alpha}{k-\lg(2^k-1)}$.*

Proof: We will use Markov's inequality. First we need to find the expectation for the number of satisfying assignments for a randomly selected formula from $\mathcal{F}_{n,m}^k$. Let $X_{n,m}^k(\varphi) = \mathcal{N}_\sigma(\varphi)$ be the random variable for φ selected randomly from $\mathcal{F}_{n,m}^k$.

$$\begin{aligned} \mathbb{E}(X_{n,m}^k) &= \sum_{\varphi \in \mathcal{F}_{n,m}^k} \mathbb{P}(\varphi) \mathcal{N}_\sigma(\varphi) \\ &= \frac{1}{(2^{\binom{n}{k}})^m} \sum_{\varphi \in \mathcal{F}_{n,m}^k} \mathcal{N}_\sigma(\varphi) = \frac{1}{(2^{\binom{n}{k}})^m} \sum_{\sigma \in \Sigma_n} \mathcal{N}_\varphi(\sigma) \\ &= \frac{1}{(2^{\binom{n}{k}})^m} \sum_{i=0}^n \binom{n}{i} \left[2^{\binom{n}{k}} - \left(\binom{i}{k} + \binom{n-i}{k} \right) \right]^m \\ &= \sum_{i=0}^n \binom{n}{i} \left(1 - \frac{\binom{i}{k} + \binom{n-i}{k}}{2^{\binom{n}{k}}} \right)^m \end{aligned}$$

Now from Markov's Inequality we have, $\mathbb{P}(X_{n,m}^k \geq 2^{\alpha n}) \leq \frac{\mathbb{E}(X_{n,m}^k)}{2^{\alpha n}}$, and to get an upper bound on the critical value for the phase transitions we will find a sufficient condition on r to make $\lim_{n \rightarrow \infty} \mathbb{P}(X_{n,rn}^k \geq 2^{\alpha n}) = 0$.

Without loss of generality we will take the limit for even n ,

$$\arg \max_i \left(1 - \frac{\binom{i}{k} + \binom{n-i}{k}}{2^{\binom{n}{k}}} \right) = \arg \min_i \frac{\binom{i}{k} + \binom{n-i}{k}}{2^{\binom{n}{k}}} = \frac{n}{2}$$

This will allow us to choose a convenient bound in the following. Also we will use the fact that

$$\lim_{j \rightarrow \infty} \frac{\binom{j}{k}}{\binom{2j}{k}} = \lim_{j \rightarrow \infty} \frac{j(j-1) \dots (j-k+1)}{2j(2j-1) \dots (2j-k+1)} = \lim_{j \rightarrow \infty} \frac{1(1-\frac{1}{j}) \dots (1-\frac{k-1}{j})}{2(2-\frac{1}{j}) \dots (2-\frac{k-1}{j})} = \frac{1}{2^k}$$

and the fact that: if $\lim_{n \rightarrow \infty} f(n) < 1$, then $\lim_{n \rightarrow \infty} (f(n))^n = 0$.

Now we are ready to find a sufficient condition on r to get the upper bound.

$$\begin{aligned} \lim_{n \rightarrow \infty} \mathbb{P}\{X_{n,rn}^k \geq 2^{\alpha n}\} &\leq \lim_{n \rightarrow \infty} \frac{\mathbb{E}(X_{n,rn}^k)}{2^{\alpha n}} \\ &\leq \lim_{n \rightarrow \infty} \frac{1}{2^{\alpha n}} \sum_{i=0}^n \binom{n}{i} \left(1 - \frac{\binom{i}{k} + \binom{n-i}{k}}{2 \binom{n}{k}}\right)^{rn} \\ &< \lim_{n \rightarrow \infty} \frac{1}{2^{\alpha n}} \sum_{i=0}^n \binom{n}{i} \left(1 - \frac{\binom{n/2}{k}}{\binom{n}{k}}\right)^{rn} \\ &= \lim_{n \rightarrow \infty} \left(2^{1-\alpha} \left(1 - \frac{\binom{n/2}{k}}{\binom{n}{k}}\right)^r\right)^n \end{aligned}$$

It follows that for values of r such that

$$\lim_{n \rightarrow \infty} \left(2^{1-\alpha} \left(1 - \frac{\binom{n/2}{k}}{\binom{n}{k}}\right)^r\right) < 1$$

will make $\lim_{n \rightarrow \infty} \mathbb{P}\{X_{n,rn}^k \geq 2^{\alpha n}\} = 0$. Now

$$\begin{aligned} \lim_{n \rightarrow \infty} \left(2^{1-\alpha} \left(1 - \frac{\binom{n/2}{k}}{\binom{n}{k}}\right)^r\right) &= \left(2^{1-\alpha} \left(1 - \lim_{n \rightarrow \infty} \frac{\binom{n/2}{k}}{\binom{n}{k}}\right)^r\right) \\ &= \left(2^{1-\alpha} \left(1 - \frac{1}{2^k}\right)^r\right) \end{aligned}$$

So we are looking for values of r such that

$$2^{1-\alpha} \left(\frac{2^k - 1}{2^k}\right)^r < 1$$

which implies an upper bound for a phase transition critical r^* of $\frac{1-\alpha}{k-\lg(2^k-1)}$. ■

Remarkably this is exactly the same result that a first moment analysis of #THRESHOLD- k -SAT($\geq 2^{\alpha n}$) would produce. So it is interesting to see if the phase transition phenomena for MONOTONE-3-SAT is the same as for 3-SAT, as this suggests.

5.2.1 Experiments for MONOTONE-3-SAT

We ran experiments to investigate the asymptotic behavior of MONOTONE-3-SAT. In effect, for each number of variables, $n = 50, 100, 150$ and 200 , we measured the fraction of random

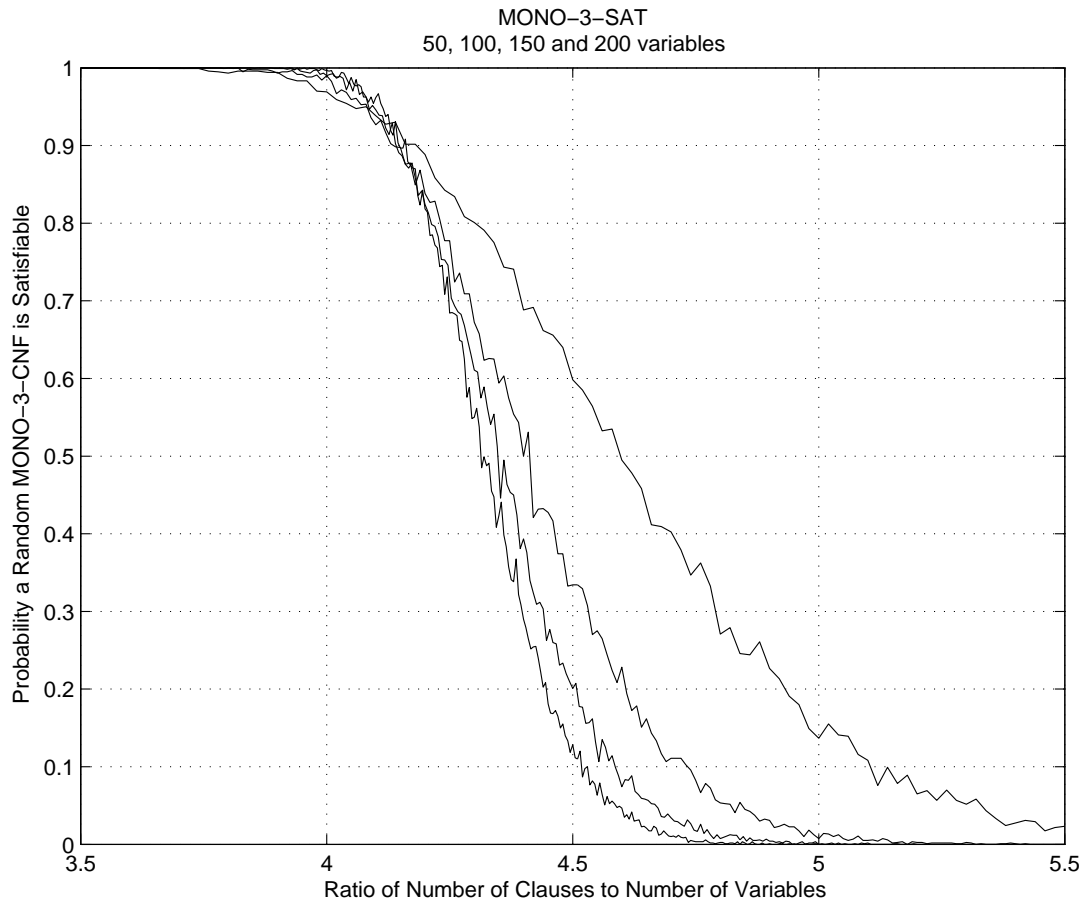


Figure 5.1:

formulas that are satisfiable in 1200 trials for all sizes of formulas with clause-to-variable ratios less than or equal to 5.5.

To form a random formula, for each disjunctive clause we uniformly drew without replacement, 3 variables, then negated all 3 variables with a probability of $1/2$. This produced a clause with literals that were either all positive or all negative. Then we used Chaff [MMZ⁺01] to test for Boolean satisfiability. The results are shown in figure 5.1. For comparison, the results for 3-SAT run on the same number of variables are shown in figure 5.2.

[h]

Table 5.1: Finite-size Scaling Fit Results for MONO-3-SAT

r_c	ν	fit
4.1000	0.6153	0.3498
4.1100	0.6351	0.3127
4.1200	0.6564	0.2813
4.1300	0.6793	0.2582
4.1400	0.7040	0.2478
4.1500	0.7306	0.2515
4.1600	0.7595	0.2732
4.1700	0.7911	0.3210
4.1800	0.8256	0.4018
4.1900	0.8636	0.5271
4.2000	0.9058	0.7099

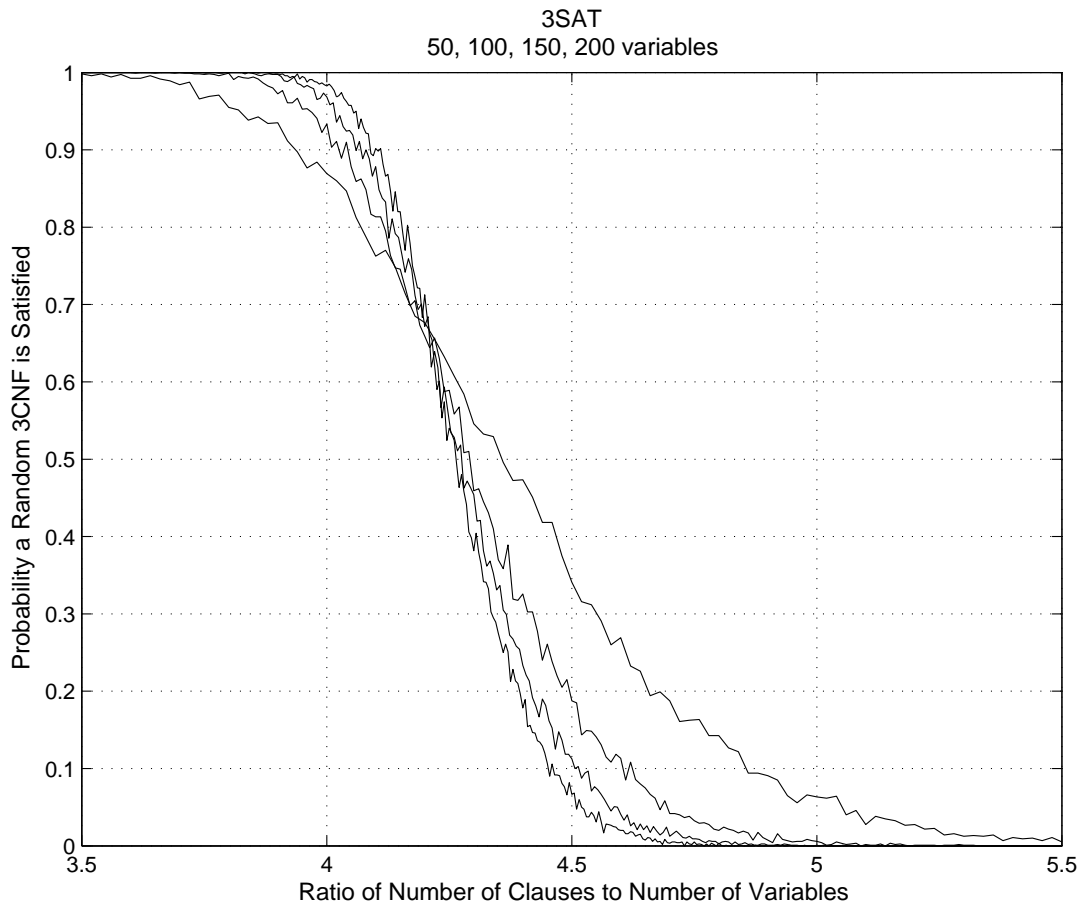


Figure 5.2:

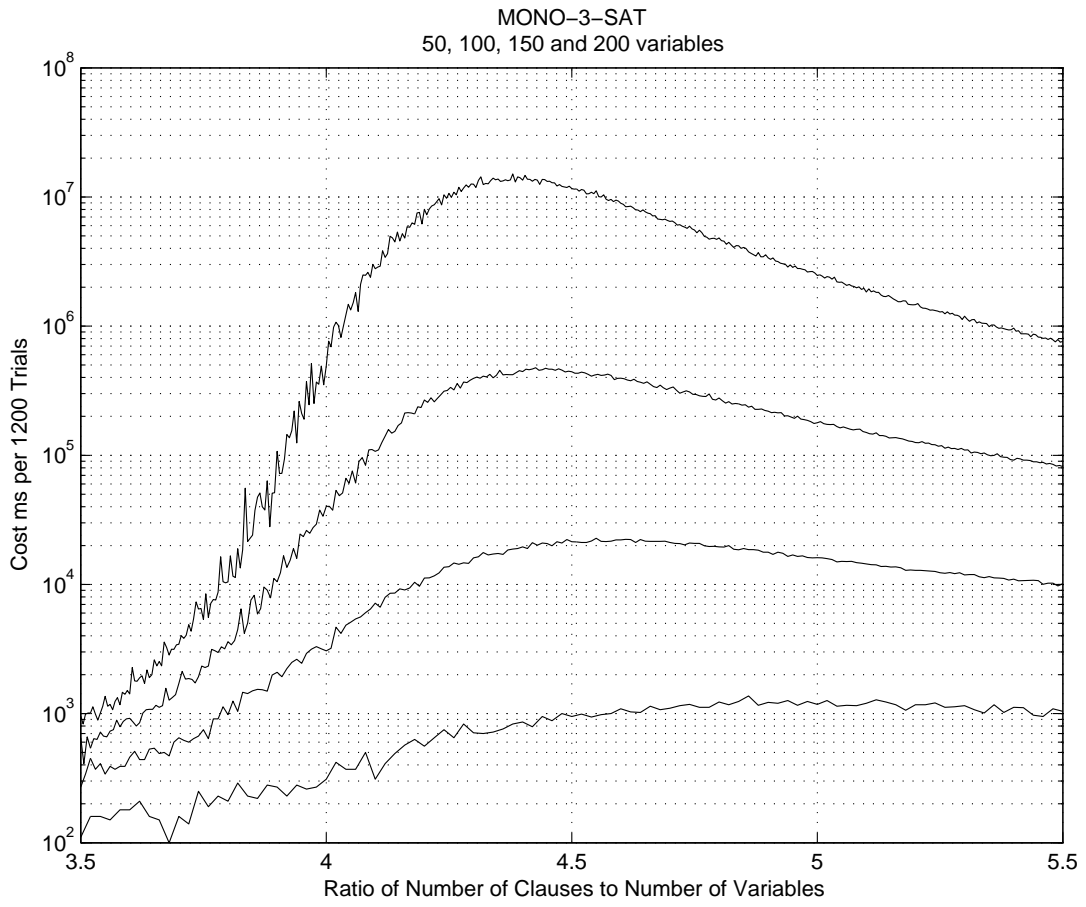


Figure 5.3:

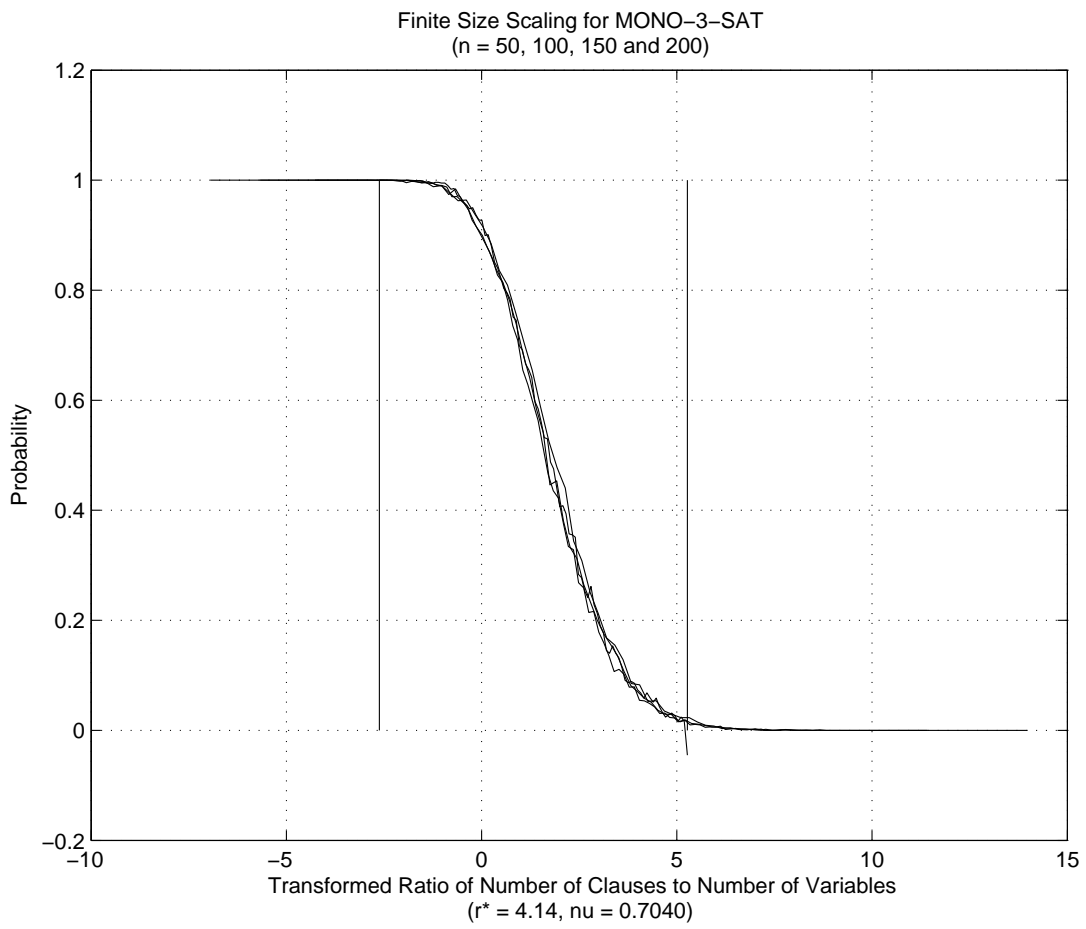


Figure 5.4:

It is clear that experimentally MONOTONE-3-SAT is very similar to 3-SAT. With a modest number of variables, it is apparent that the probability curves steepen with increasing n and a phase transition appears to occur near the same critical ratio as for 3-SAT. Finite-size scaling on the MONOTONE-3-SAT data indicates about 4.14 and on the 3-SAT data about 4.2. The finite-size scaling method we use optimizes both the estimates of the critical ratio r^* and the exponent ν by regression rather than just visually estimating r^* [BK03]. Given the size of these data sets this difference is very likely not significant and it is reasonable to conjecture that the critical ratio is the same for these two problems. Analytically, this is an open question. The experimental results also show some interesting differences with 3-SAT, for the same values of n , the slopes of the probability curves at the crossover point are less steep and the probability at which the curves appear to intersect is much higher, 0.92 rather than about 0.6. We also recorded the algorithmic costs for the MONOTONE-3-SAT runs and observed the same “easy-hard-easy” pattern well known for 3-SAT.

5.3 Generalized Satisfiability

Individual satisfiability variant problems can be separately analyzed as above. And in general, the analysis is not necessarily as simple as in the case of 3-SAT. We found that even though the different variants may introduce their own peculiar complexities, a general approach can handle much of the analysis and serve to isolate the area that requires individual treatment for each of the problems. Part of Schaefer’s work on generalized Boolean satisfiability provides a ready framework for this approach when the concepts related to random formulas are added.

Definition 5.3.1: Let S be a finite set of relations, $\{R_1, R_2, R_3, \dots, R_N\}$, of various arities, let $\widehat{S} = \{\widehat{R}_1, \widehat{R}_2, \widehat{R}_3, \dots, \widehat{R}_N\}$ be a set of corresponding relation symbols with matching arities and let $X = \{x_1, x_2, x_3, \dots, x_n\}$ be a set of n variables.

A CNF(S)-clause is a formula of the form $\widehat{R}(v_1, v_2, v_3, \dots, v_k)$ where \widehat{R} is a relational

symbol in \widehat{S} and $v_1, v_2, v_3, \dots, v_k$ are variables in X .

An m -clause CNF(S)-formula, $C_1 \wedge C_2 \wedge C_3 \wedge \dots \wedge C_m$, is a conjunction of m CNF(S)-clauses. The semantics of CNF(S)-formulas are defined in a standard way, assuming the domain of each of the variables to be $\{0, 1\}$ and the interpretation of each relational symbol \widehat{R}_i to be the corresponding relation R_i in S .

The decision problem SAT(S) is the following: given a CNF(S)-formula φ , is it satisfiable? In other words, is there a truth assignment of its variables that will make all of its clauses true?

An m -clause random CNF(S)-formula over X is an m -clause CNF(S)-formula where each of its clauses has been generated by a uniformly random selection, with replacement, of its relational symbol from \widehat{S} and then each of its variables, in order up to its arity, by a uniformly random selection, without replacement, from X . The sample space of m -clause random CNF(S)-formulas over X is denoted by $F_S(n, m)$.

Given a real, r , greater than zero and a sequence of spaces, $F_S(n, \lceil nr \rceil)$. Let $\varphi_{n,r}$ be a sequence of random formulas drawn from the spaces, then we say that there is a phase transition in the probability of $\varphi_{n,r}$ being satisfiable if there exists a critical value r^* such that,

$$\lim_{n \rightarrow \infty} \text{P}(\varphi_{n,r} \text{ is satisfiable}) = \begin{cases} 1 & \text{if } r < r^* \\ 0 & \text{if } r > r^* \end{cases}$$

■

This definition simply extends Schaefer's framework for generalized satisfiability problems to include the concepts of random formulas and phase transitions. We will not make use of his classification scheme for relations. Phase transitions are known to occur both analytically and experimentally in both P and NP-complete Boolean satisfiability problems.

5.3.1 Upper bounds for Generalized Satisfiability

In the following we will make use of the concept of power mean.

Definition 5.3.2: Let, $A = \{a_i : i = 1, 2, 3, \dots, n\}$, where the a_i are non-negative reals. Then

$$\mathfrak{M}_p(A) = \left(\frac{1}{n} \sum_{i=1}^n a_i^p \right)^{\frac{1}{p}}$$

is the p power mean of A [HLP52]. ■

Theorem 5.3.3: *If a critical r^* exists for SAT(S), then it must be less than $\frac{1}{\lg\left(\frac{N_c}{\max N_T(\sigma)}\right)}$.*

Proof: Let SAT(S) be a generalized satisfiability decision problem. Let $F_S(n)$ be the associated space of $CNF(S)$ -formulas over its n variables and $F_S(n, m)$ the space of m clause ones. Define indicator functions I_σ for each possible assignment, σ , of values to the variables as follows:

$$I_\sigma : F_S(n) \rightarrow \{0, 1\} \text{ where}$$

$$I_\sigma(\varphi) = \begin{cases} 1 & \text{if } \sigma \text{ satisfies } \varphi \\ 0 & \text{otherwise} \end{cases},$$

And let, $N(\varphi) = \sum_\sigma I_\sigma(\varphi)$. Then $I_\sigma(\varphi)$ and $N(\varphi)$ are random variables and it follows that

$$\mathbb{E}(N(\varphi)) = \sum_\sigma \mathbb{E}(I_\sigma(\varphi)).$$

Let N_c be total number of possible kinds of clauses that can be in a random formula and let $N_T(\sigma)$ be a function that indicates the number of those clauses that σ satisfies.

Then for the space $F_S(n, m)$,

$$\mathbb{E}(I_\sigma(\varphi)) = \sum_\varphi \mathbb{P}(\varphi) I_\sigma(\varphi) = \frac{1}{(N_c)^m} \sum_\varphi I_\sigma(\varphi) = \frac{N_T(\sigma)^m}{(N_c)^m}$$

and

$$\begin{aligned}
\mathbf{E}(N(\varphi)) &= \sum_{\sigma} \left(\frac{N_T(\sigma)}{N_c} \right)^m = 2^n \frac{1}{2^n} \sum_{\sigma} \left(\frac{N_T(\sigma)}{N_c} \right)^m \\
&= 2^n \left(\left(\frac{1}{2^n} \sum_{\sigma} \left(\frac{N_T(\sigma)}{N_c} \right)^m \right)^{\frac{1}{m}} \right)^m \\
&= 2^n \left(\mathfrak{M}_m \left(\left\{ \frac{N_T(\sigma)}{N_c} \mid \sigma \in \Sigma_n \right\} \right) \right)^m \\
&= \left(2 \left(\mathfrak{M}_m \left(\left\{ \frac{N_T(\sigma)}{N_c} \mid \sigma \in \Sigma_n \right\} \right) \right)^r \right)^n \quad \text{where } r = \frac{m}{n}.
\end{aligned}$$

We are looking for sufficient conditions for this quantity to tend to zero when n tends to infinity.

Note that if

$$\lim_{n \rightarrow \infty} 2 \left(\mathfrak{M}_{nr} \left(\left\{ \frac{N_T(\sigma)}{N_c} \mid \sigma \in \Sigma_n \right\} \right) \right)^r < 1,$$

then

$$\lim_{n \rightarrow \infty} \mathbf{E}(X_{N(\varphi)}) \rightarrow 0$$

Now for all n ,

$$\mathfrak{M}_{nr} \left(\left\{ \frac{N_T(\sigma)}{N_c} \mid \sigma \in \Sigma_n \right\} \right) \leq \frac{\max N_T(\sigma)}{N_c}$$

and

$$\lim_{n \rightarrow \infty} \mathfrak{M}_{nr} \left(\left\{ \frac{N_T(\sigma)}{N_c} \mid \sigma \in \Sigma_n \right\} \right) \leq \frac{\max N_T(\sigma)}{N_c}$$

So a sufficient condition for the expectation to tend to zero is for r to satisfy the inequality

$$\begin{aligned}
2 \left(\frac{\max N_T(\sigma)}{N_c} \right)^r &< 1 \\
r &< \frac{1}{\lg \left(\frac{N_c}{\max N_T(\sigma)} \right)}
\end{aligned}$$

■

5.4 1-IN-3-SAT Bound and Experiments

5.4.1 Upper Bound

Definition 5.4.1: The 1-IN-3-SAT decision problem is a version of 3-SAT where the semantics of the clauses is changed to require that *exactly one* literal in each clause be true rather than at least one. It has as input a 3CNF formula in which all clauses are 1-in-3 relations and has as output the answer YES if there exists at least one assignment of values to the variables in the formula, such that all of the clauses in the formula are satisfied, and the answer NO, if for all possible assignments, there are always one or more clauses which are not satisfied. Where the 1-in-3 relations are:

$$R_{+++} = \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$$

$$R_{-++} = \{(0, 0, 0), (1, 1, 0), (1, 0, 1)\}$$

$$R_{+--} = \{(1, 1, 0), (0, 0, 0), (0, 1, 1)\}$$

$$R_{++-} = \{(1, 0, 1), (0, 1, 1), (0, 0, 0)\}$$

$$R_{--+} = \{(0, 1, 0), (1, 0, 0), (1, 1, 1)\}$$

$$R_{-+-} = \{(0, 0, 1), (1, 1, 1), (1, 0, 0)\}$$

$$R_{+--} = \{(1, 1, 1), (0, 0, 1), (0, 1, 0)\}$$

$$R_{---} = \{(0, 1, 1), (1, 0, 1), (1, 1, 0)\}$$

■

Note, relations have variables as input rather than literals. The subscripts in the above definition correspond to all the possible patterns of assertion and negation of the input variables. Also note that in the usual application of Schaefer's dichotomy theorem, only R_{+++} , R_{-++} , R_{--+} and R_{---} would be considered, since given these, the remaining relations can be realized by permuting input variables and the Schaefer classes of relations are invariant with respect to permutations of the variables. For

this study, all of the permutations must be considered because the standard way of generating random formulas produces all of them.

The 1-IN-3-SAT problem is noteworthy as the first NP-complete problem to have an analytically proven phase transition. The location of the critical clause-to-variable ratio is $1/3$ [ACIM01]. For this study, the fact that this critical ratio has been established analytically is useful because it allows comparing this true value with the general upper bound analysis, the results from the experiments reported here and the results of the finite-size scaling procedures.

To find the upper bound that can be established with a first moment argument, we note that for all assignments, σ , and all possible combinations of three variables, the assignment makes three 1-in-3 type relational clauses *true* and five *false*. So for all σ ,

$$\frac{N_T(\sigma)}{N_c} = \frac{3 \binom{n}{3}}{2^3 \binom{n}{3}} = \frac{3}{8}$$

and it follows that

$$r < \frac{1}{\lg \left(\frac{N_c}{\max N_T(\sigma)} \right)} \approx 0.7067$$

5.4.2 Experiments

We ran experiments to investigate the asymptotic behavior of 1-IN-3-SAT. In effect, for each number of variables, $n = 100, 200, 300, 400, 500, 600, 700$ and 800 , we measured the fraction of random formulas that are satisfiable in 1200 trials for all sizes of formulas with clause-to-variable ratios less than or equal to one.

To form a random formula, for each relational clause we uniformly drew without replacement, 3 variables, then negated each variable with a probability of $1/2$ yielding a triple of literals, say, u, v and w , then we formed a Boolean formula equivalent for the desired relational clause, namely,

$$(u \vee v \vee w) \wedge (u \vee \neg v \vee \neg w) \wedge (\neg u \vee v \vee \neg w) \wedge (\neg u \vee \neg v \vee w) \wedge (\neg u \vee \neg v \vee \neg w)$$

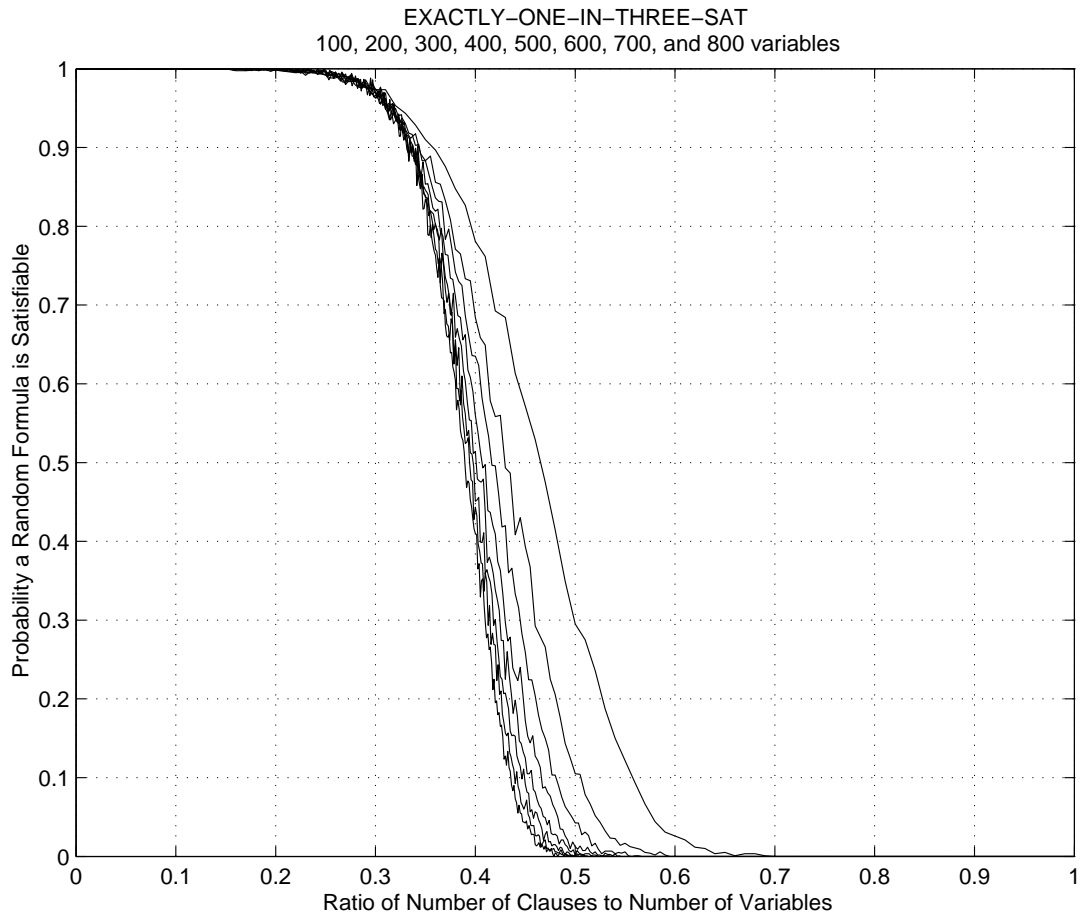


Figure 5.5:

So, in effect, equivalent CNF Boolean formulas were formed on the fly corresponding to the randomly selected 1-IN-3 clauses. After this procedure, we used Chaff to test for Boolean satisfiability. The power of Chaff made it possible to work with large numbers of variables on modest computer hardware. With respect to time, this approach does not necessarily produce the most efficient algorithm for every case of Schaefer generalized satisfiability. However, the structural phase transition picture is the same for any sound and complete algorithm. The results are shown in figure 5.5. Finite-size scaling estimated $r^* = 0.32$ and $\nu = 0.3086$.

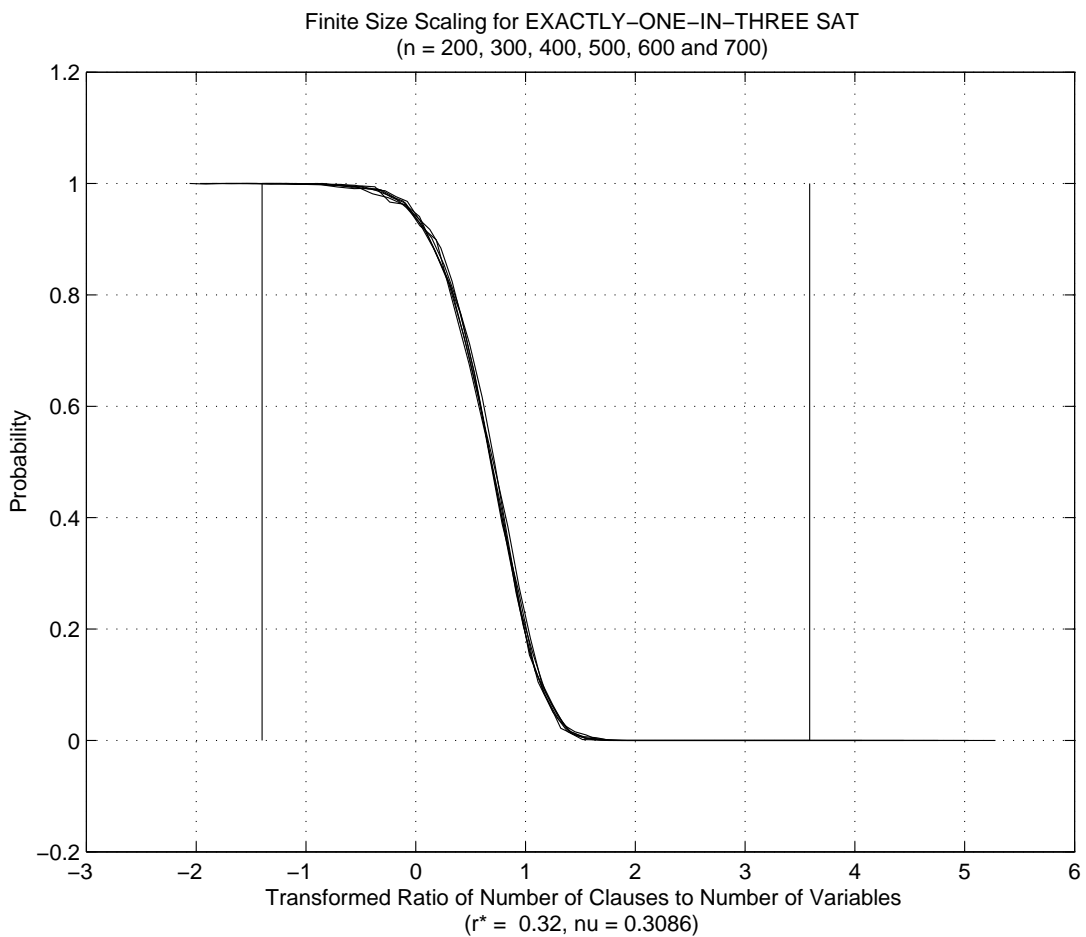


Figure 5.6:

Table 5.2: Finite-size Scaling Fit Results for 1-IN-3-SAT

r_c	ν	fit
0.2800	0.2135	0.0908
0.2900	0.2313	0.0803
0.3000	0.2524	0.0707
0.3100	0.2777	0.0632
<i>0.3200</i>	<i>0.3086</i>	<i>0.0606</i>
0.3300	0.3475	0.0674
0.3400	0.3978	0.0937
0.3500	0.4654	0.1650
0.3600	0.5617	0.3449
0.3700	0.7110	0.8371
0.3800	0.9797	2.5540

5.5 Positive 1-IN-3-SAT Bound and Experiments

5.5.1 Upper Bound

Definition 5.5.1: The POS-1-IN-3-SAT problem is a decision problem which has as input a CNF(σ) formula in which all clauses are positive 1-in-3 relations and which has as output YES if there exists at least one assignment of values to the variables in the formula such that all of the clauses in the formula are satisfied, and the answer is NO, if for all possible assignments, there are always one or more clauses which are not satisfied. Where the positive 1-in-3 relation is:

$$R_{+++} = \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$$

■

To find the upper bound that can be established with a first moment argument, we note that for all assignments, σ , and all possible combinations of three variables, the number of positive 1-in-3 type relational clauses that an assignment satisfies varies with the number of variables made *true* by the assignment. Let t_σ be the number of variables assigned *true* by σ , then for all σ ,

$$\frac{N_T(\sigma)}{N_c} = \frac{\binom{t_\sigma}{1} \binom{n-t_\sigma}{2}}{\binom{n}{3}}$$

Now as $n \rightarrow \infty$, in the limit

$$\arg \max_{\sigma} \binom{t_\sigma}{1} \binom{n-t_\sigma}{2} = \arg \max_{\sigma} t_\sigma (n-t_\sigma)^2$$

The roots of this cubic are $n/3$ and n and clearly the maximum occurs for $n/3$. Figure 5.7 shows the variation in number of *true* clauses when $n = 300$.

So we have

$$\frac{N_c}{\max N_T(\sigma)} = \frac{\binom{n}{3}}{\frac{n}{3} \binom{2n/3}{2}} = \frac{9}{4}$$

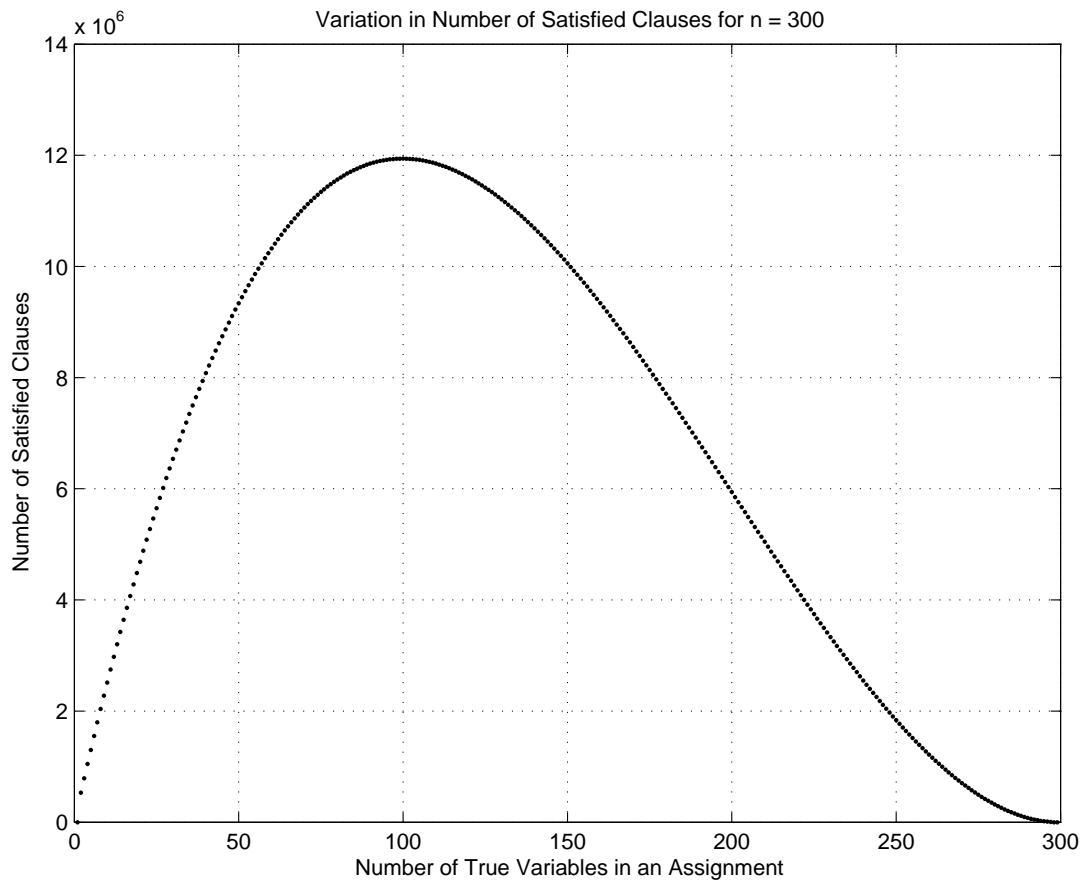


Figure 5.7:

and

$$r < \frac{1}{\lg\left(\frac{N_s}{\max N_T(\sigma)}\right)} \cong 0.8548$$

5.5.2 Experiments

We ran experiments to investigate the asymptotic behavior of POS-1-IN-3-SAT. In effect, for each number of variables, $n = 100, 200, 300, 400, 500, 600$ and 700 , we measured the fraction of random formulas that are satisfiable in 1200 trials for all sizes of formulas with clause-to-variable ratios less than or equal to one.

To form a random formula, for each relational clause we uniformly drew, without replacement, 3 variables. Since only positive literals are considered for this case there is no negation process as there was in 1-IN-3-SAT. Instead, the 3 selected variables directly yield a triple of positive literals, say, u, v and w . Then we formed a Boolean formula equivalent for the desired relational clause, namely,

$$(u \vee v \vee w) \wedge (u \vee \neg v \vee \neg w) \wedge (\neg u \vee v \vee \neg w) \wedge (\neg u \vee \neg v \vee w) \wedge (\neg u \vee \neg v \vee \neg w)$$

After this, the procedure was the same as that done for 1-IN-3-SAT. The results are shown in figure 5.8. The finite-size scaling estimates were $r^* = 0.62$ and $\nu = 0.5815$.

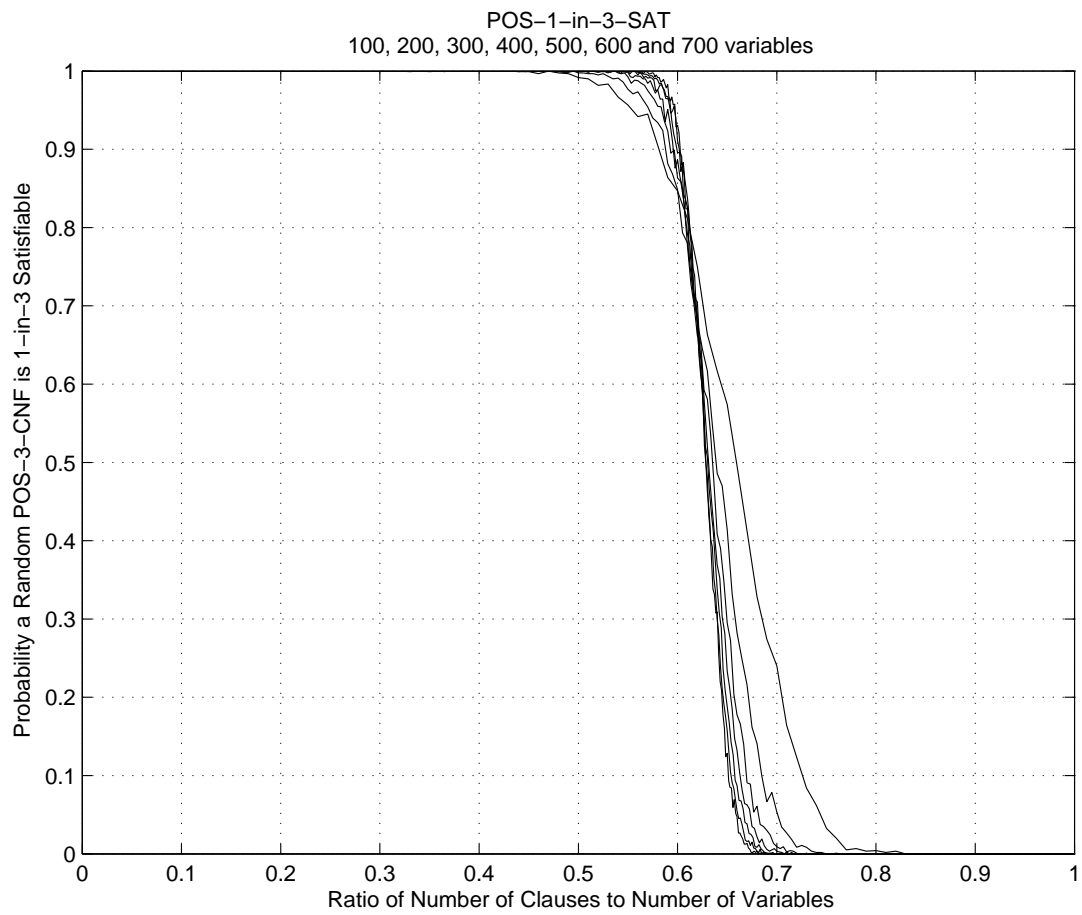


Figure 5.8:

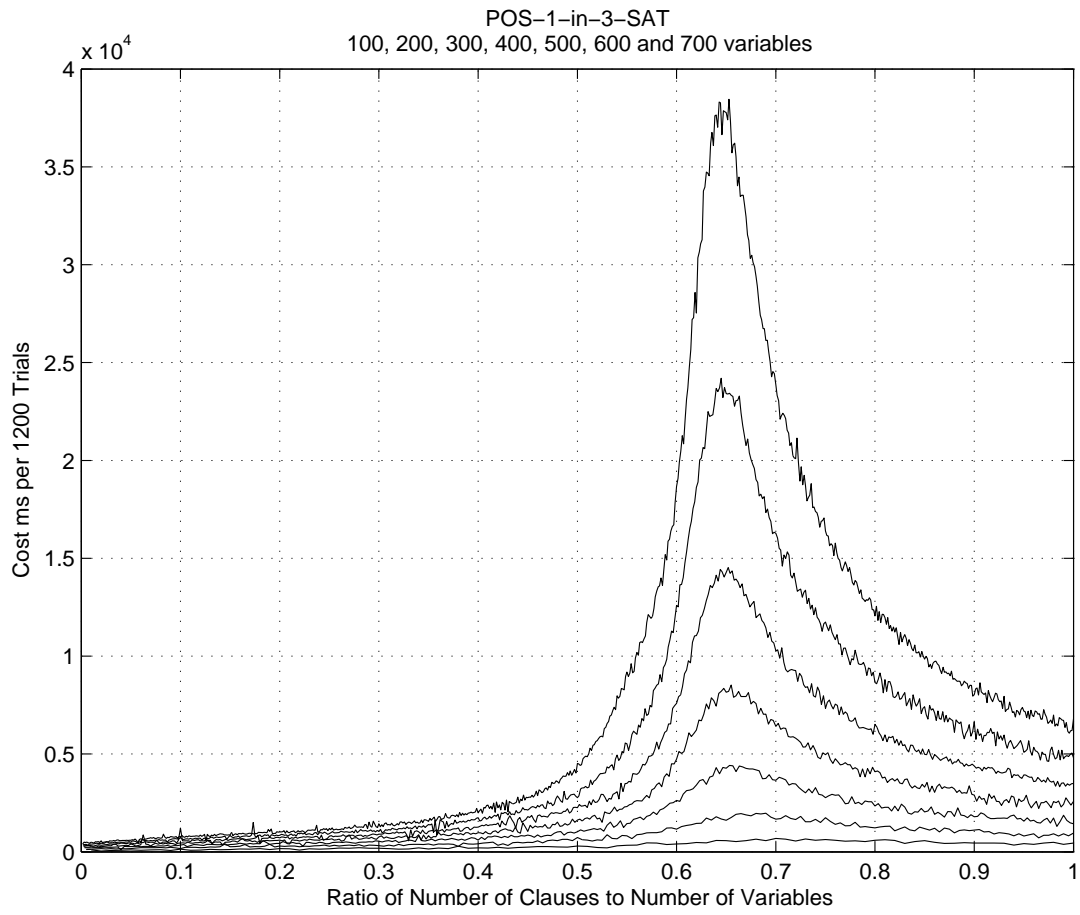


Figure 5.9:

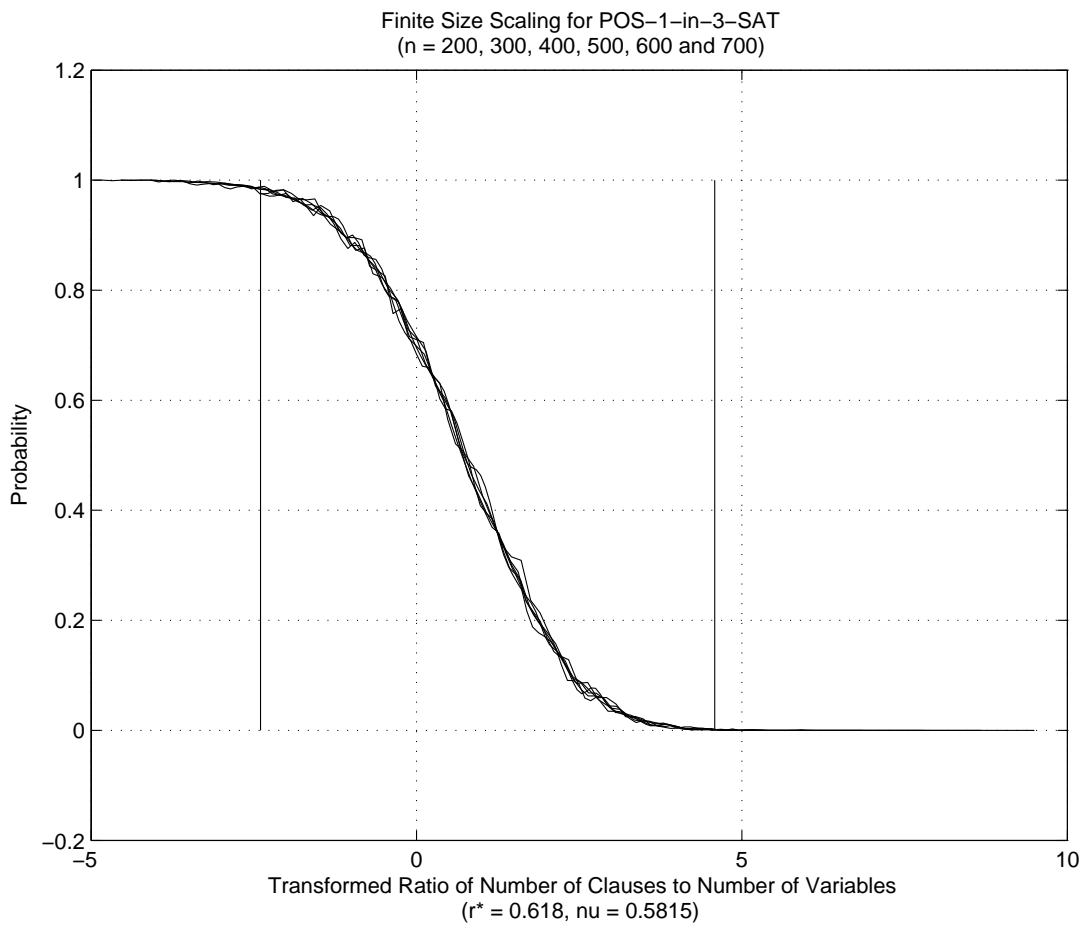


Figure 5.10:

Table 5.3: Finite-size Scaling Fit Results for POS-1-IN-3-SAT

r_c	ν	fit
0.6100	0.3734	0.8302
0.6110	0.3907	0.7353
0.6120	0.4098	0.6388
0.6130	0.4309	0.5408
0.6140	0.4543	0.4475
0.6150	0.4805	0.3639
0.6160	0.5099	0.2961
0.6170	0.5433	0.2538
<i>0.6180</i>	<i>0.5815</i>	<i>0.2533</i>
0.6190	0.6258	0.3187
0.6200	0.6776	0.4937
0.6210	0.7394	0.8521
0.6220	0.8144	1.5283
0.6230	0.9076	2.7510
0.6240	1.0276	4.9455
0.6250	1.1893	8.9142

In summary, in this part we used the first moment method to find an upper bound for the MONOTONE- k -SAT phase transition. The procedure is somewhat more complex than the usual simple one for k -SAT. Remarkably, the bound turns out to be the same as the one for k -SAT. We ran experiments to compare the apparent phase transitions for the case with $k = 3$ and they are indeed very similar. The experimental results with finite-size scaling indicate critical ratios of 4.2 and 4.14 for 3-SAT and MONOTONE-3-SAT respectively. Given the small numbers of variables used in the runs this small experimental difference may not reflect a real difference in the ultimate asymptotic behavior. There are definite differences in the steepness of the transition for comparable n and the probability level at which the curves appear to intersect is much higher for MONOTONE-3-SAT than for 3-SAT. In order to explore other variants of Boolean satisfiability, we developed a generalized method for determining first moment upper bounds for phase transitions in generalized random problems in Schaefer's framework for generalized satisfiability. We demonstrated the utility of the technique with analysis and experiments for 1-IN-3-SAT and POS-1-IN-3-SAT. The technique produces tighter bounds than the previously reported uniform approach of Creignou and Daudé [CD03]. The comparisons for 3-SAT, MONOTONE-3-SAT, 1-IN-3-SAT and POS-1-IN-3-SAT are respectively 44.36 vs 5.19, 11.08 vs 5.19, 44.36 vs 0.71, and 5.14 vs 0.85. It should be noted that their goal was not to produce tight bounds but to prove that all of the non-trivial generalized satisfiability problems with fixed arity have the same scale factor with respect to phase transitions.

There are two promising future directions for this work. It appears possible to incorporate the sharpened first moment method of Kirousis, et al [KKKS98] into the generalized technique to further tighten the attainable bounds. An open question for analysis is: are the locations of the phase transitions for MONOTONE- k -SAT and k -SAT the same?

Part VI

Summary and Conclusions

Chapter 6

Summary and Conclusions

During the past decade, there has been an intensive investigation of phase transitions in Boolean satisfiability problems. Most of this work has been related to the k -SAT decision problems over spaces of random k -CNF formulas parameterized with respect to number of variables, number of clauses and size of clauses. The interest in these problems has been motivated primarily by two facts. First, that there is a dramatic change in the asymptotic satisfiability of random formulas with respect to a simple critical parameter, the ratio of the number of clauses to the number of variables. Second, that there appears to be a corresponding variation in the average time it takes for the best known algorithms to solve the formulas. Extensive experiments suggest that the variation is due to the density of hard instances being highest in the neighborhood of the location of the phase transition.

Following the strategy that useful insights into a difficult problem can sometimes be gained from studying generalizations of it, we explored three different kinds of generalizations of satisfiability with the combined goals: of extending the understanding of how the structural phase transition emerges and how its emergence affects the behavior of algorithms. We explored three different generalizations specified as three different variations in the definition of the SAT decision problem: Threshold Counting SAT, a decision problem that allows specification of a threshold value for the number of satisfying

assignments that must exist; Bounded SAT, a decision problem that allows specification of a bound for the maximum number of clauses that may be violated; and Schaefer's Generalized SAT, a decision problem that allows specification of permissible clause types which can include more than just simple disjunctions.

In studying threshold counting SAT we discovered a family of PP-complete decision problems that exhibit threshold behavior on the same scale as SAT and experimentally show similar phase transitions. These are the #THRESHOLD- k -SAT($\geq 2^{\alpha n}$) decision problems, where n is the number of variables and, $0 \leq \alpha < 1$, is a real. If $\alpha = 0$, this is simply the k -SAT problem. For values of α greater than 0, these decision problems experimentally show phase transition behavior similar to k -SAT but with the location of the critical ratio occurring at progressively smaller values as α approaches 1. In observing the behavior of the family with respect to the location of the phase transition and the algorithmic cost, we discovered that there is a tight relationship between the threshold decision problem and its associated counting problem that leads to the critical ratio not always corresponding to the location of peak algorithmic cost.

Future directions for study of these problems include: obtaining exact analytical results for the existence of the phase transition, finding tighter analytical bounds and characterizing the transition as sharp versus coarse in Friedgut's framework. Exact results can be expected to be at least as elusive as they have been for the simpler problem k -SAT. None of the techniques that have yielded better bounds for k -SAT appear to be useful for the #THRESHOLD- k -SAT($\geq 2^{\alpha n}$) problem. Experimental results indicate that as α increases the slope of the transition becomes steeper for any given n which, if true, would imply that since the threshold for k -SAT is sharp, the threshold for #THRESHOLD- k -SAT($\geq 2^{\alpha n}$) must also be sharp. This experimental observation is an open question for analysis.

In studying Zhang's bounded satisfaction problems, k SAT(b), we discovered a new type of emergence of the phase transition which clearly shows that the peak cost point is not tied to the phase transition location but to the cross-over point. Actually, the relation to the cross-over point was first

pointed out in 1992 [MSL92]. Evidently since the cross-over points for the k -SAT problem are close to the phase transition, many refer to the peak as being related to the phase transition location. Previously reported experiments for this problem had suggested that there are separate phase transition locations for each value of the bound. Our experimental results show that asymptotically the locations of the transitions for different bounds appear to move toward the location of the transition for the unbounded problem. Analytically, we showed that they have the same Markov upper bound as the unbounded problem regardless of the value of the bound, and in the case of $2SAT(b)$, we showed the bounded versions have exactly the same phase transition location.

There are two interesting open questions related to bounded satisfiability. One, can it be analytically shown that k -SAT and k -SAT(B) for any b have the same asymptotic values for the threshold location? Second, is there a variation of this problem that will produce a family of separate locations for the transition as is the case for $\#THRESHOLD-k$ -SAT($\geq 2^{\alpha n}$)? Preliminary investigation suggests that specifying the bound as a percentage of clauses would produce this result.

In studying variants of satisfiability in Schaefer's generalized SAT framework, we found that $MONOTONE-k$ -SAT appears to have a phase transition very near, or at the same point as, k -SAT and furthermore, that the analytical Markov upper bounds for the two problems are the same. We developed a uniform method for determining the upper bounds for phase transitions in random Schaefer generalized SAT problems that produces much tighter bounds than the previously published uniform technique. The fact that the analytical result is not correlated in any way with the polynomial time Schaefer classes of relations suggests that, in these problems, complexity and phase transitions are not related. This is not surprising given that it is well known that k -SAT, both for $k = 2$ and $k = 3$, have sharp thresholds, but one is decidable in P time and the other is NP-complete.

An open question in this area is: are the locations of the phase transitions, if they exist, the same for $MONOTONE-k$ -SAT and k -SAT? Another future direction for analysis that looks promising is to tighten the uniform upper bounds method even more by incorporating the sharpened first

moment method of Kirousis, et al.

A major overall open question for phase transition research is: are there identifiable structural characteristics that will imply the existence of a phase transition. Unlike the concepts of “threshold”, “coarse threshold”, and “sharp threshold” at present there are not any known analytical results that tie some *structural* condition, such as monotonicity, non-local-approximability, etc., to the existence of a phase transition.

Bibliography

- [Ach00] D. Achlioptas. Setting two variables at a time yields a new lower bound for random 3-SAT. In *Proc. 32th ACM Symp. on Theory of Computing*, pages 28–37, 2000.
- [ACIM01] Dimitris Achlioptas, Arthur Chtcherba, Gabriel Istrate, and Cristopher Moore. The phase transition in 1-in-k SAT and NAE 3-SAT. In *Symposium on Discrete Algorithms (SODA) 2001*, pages 721–722, 2001.
- [AM02] D. Achlioptas and C. Moore. The asymptotic order of the random k-sat threshold, 2002.
- [And89] Ian Anderson. *Combinatorics of Finite Sets*. Oxford University Press, Oxford, UK, 1989.
- [Ang80] D. Angluin. On counting problems and the polynomial-time hierarchy. *Theoretical Computer Science*, 12:161–173, 1980.
- [Bak95] A. Baker. Intelligent backtracking on the hardest constraint problems. Technical report, CIRL, University of Oregon, February 1995.
- [BBC⁺01] Béla Bollobás, Christian Borgs, Jennifer T. Chayes, Jeong Han Kim, and David B. Wilson. The scaling window of the 2-SAT transition. *Random Structures and Algorithms*, 18(3):201–256, 2001.
- [BDK01] Delbert D. Bailey, Victor Dalmau, and Phokion G. Kolaitis. Phase transitions of PP-complete satisfiability problems. *Proc. of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)*, pages 183–189, 2001.
- [BFU93] A. Z. Broder, A. M. Frize, and E. Upfal. On the satisfiability and maximum satisfiability of random 3-CNF formulas. In *Proc. 4th Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 322–330, 1993.
- [BK03] Delbert D. Bailey and Phokion G. Kolaitis. Phase transitions of bounded satisfiability problems. *Proc. of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 1187–1193, 2003.
- [BL99] E. Birnbaum and E.L. Lozinskii. The good old Davis-Putnam procedure helps counting models. *Journal of Artificial Intelligence Research*, 10:457–477, 1999.
- [Bol86] B. Bollobás. *Combinatorics*. Cambridge University Press, Cambridge, UK, 1986.

- [BP00] R.J. Bayardo and J.D. Pehoushek. Counting models using connected components. In *17th Nat'l Conf. on Artificial Intelligence*, pages 157–162, 2000.
- [BT87] B. Bollobás and A. Thomason. Threshold functions. *Combinatorica*, 7:35–38, 1987.
- [By194] T. Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69:161–204, 1994.
- [By196] T. Bylander. A probabilistic analysis of propositional STRIPS planning. *Artificial Intelligence*, 81:241–271, 1996.
- [CD03] Nadia Creignou and Hervé Daudé. Generalized satisfiability problems: minimal elements and phase transitions. *Theoretical Computer Science*, 302(1–3):417–430, 2003.
- [CF86] M. Chao and J. V. Franco. Probabilistic analysis of two heuristics for the 3-satisfiability problem. *SIAM J. on Computing*, 15(4):1106–1118, 1986.
- [CF90] M. Chao and J. V. Franco. Probabilistic analysis of a generalization of the unit-clause literal selection heuristics for the k -satisfiability problem. *Information Sciences*, 51:289–314, 1990.
- [CGS97] 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.
- [CKT91] P. Cheeseman, B. Kanefsky, and W. M. Taylor. Where the really hard problems are. In *Proc. 12th Int'l Joint Conf. on Artificial Intelligence (IJCAI '91)*, pages 331–337, 1991.
- [Coo71] S. A. Cook. The complexity of theorem-proving procedures. *Proc. of 3rd Annual ACM Symposium on the Theory of Computing*, pages 151–158, 1971.
- [CR92a] 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.
- [CR92b] V. Chvátal and B. Reed. Mick gets some (the odds are on his side). In *Proceedings 33rd Symposium on the Foundations of Computer Science, IEEE Press*, pages 620–627, 1992.
- [DBM00] O. Dubois, Y. Boufkhad, and J. Mandler. Typical random 3-SAT formulae and the satisfiability threshold. In *Proc. 11th Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 126–127, 2000.
- [ER59] P. Erdős and A. Rényi. On random graphs I. *Publ. Math. Debrecen*, pages 290–297, 1959.
- [ER60] P. Erdős and A. Rényi. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, pages 17–61, 1960.
- [Erd47] P. Erdős. Some remarks on the theory of graphs”” *Bull. Amer. Math. Soc.*, 53:292–294, 1947.
- [Fer92a] W. Fernandez de la Vega. On random 2-SAT. Manuscript, 1992.
- [Fer92b] W. Fernandez de la Vega. On random 2-SAT. Manuscript, 1992.

- [FP83] J. V. Franco and M. Paul. Probabilistic analysis of the Davis-Putnam Procedure for solving the satisfiability problem". *Discrete Applied Math.*, 5:77–87, 1983.
- [Fri99] E. Friedgut. Sharp threshold of graph properties and the k -SAT problem. *J. Amer. Math. Soc.*, 12:1917–1054, 1999.
- [FS96] A. Frieze and S. Suen. Analysis of two simple heuristics for random instances of k -SAT. *J. of Algorithms*, 20(2):312–355, 1996.
- [Gil77] J. Gill. Computational complexity of probabilistic Turing machines. *SIAM J. Comput.*, 6(4):675–695, 1977.
- [GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, NY, 1979.
- [Goe96a] A. Goerdt. A threshold for unsatisfiability. *J. Comput. System Sci.*, 53(3):469–486, 1996.
- [Goe96b] A. Goerdt. A threshold for unsatisfiability. *Journal of Computer and System Sciences*, 53(3):469–486, 1996.
- [Goe96c] Andreas Goerdt. A threshold for unsatisfiability. *Journal of Computer and System Sciences*, 53:469–486, 1996.
- [Gol79] Goldberg. On the complexity of the satisfiability problem. Courant computer science report no. 16, New York University, New York, NY, 1979.
- [GW94] Ian Gent and Toby Walsh. Easy problems are sometimes hard. *Artificial Intelligence*, 70:335–345, 1994.
- [GW96a] Ian Gent and Toby Walsh. Phase transitions and annealed theories: number partitioning. *Proc. of 12th European Conf. on Artificial Intelligence, ECAI-96*, pages 170–174, 1996.
- [GW96b] Ian Gent and Toby Walsh. The TSP phase transition. *Artificial Intelligence*, 88(1–2):349–358, 1996.
- [GW99] I.P. Gent and T. Walsh. Beyond NP: the QSAT phase transition. In *Proc. 16th National Conference on Artificial Intelligence*, pages 653–658, 1999.
- [HLP52] G. H. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities, 2nd Ed.* Cambridge University Press, Cambridge, Great Britain, 1952.
- [HW94] T. Hogg and C. P. Williams. The hardest constraint problems: A double phase transition. *Artificial Intelligence*, 69(1-2):359–377, 1994.
- [Ist00] Gabriel Istrate. Computational complexity and phase transitions. *Proceedings of 15th Annual IEEE Conference on Computational Complexity (CoCo'00), July 04-07, Florence Italy, 2000*, pages 104–115, 2000.
- [JLR00] Svante Janson, Tomasz Łuczak, and Andrzej Ruciński. *Random Graphs*. Wiley-Interscience, New York, N.Y., 2000.
- [JSV00] 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.

- [KKKS98] L.M. Kirousis, E. Kranakis, D. Krizanc, and Y.C. Stamatiou. Approximating the unsatisfiability threshold of random formulas. *Random Structures and Algorithms*, 12:253–269, 1998.
- [KKL02] A. C. Kaporis, L. M. Kirousis, and E. G. Lalas. The probabilistic analysis of a greedy satisfiability algorithm. Algorithms-ESA2002, Proceedings of the Tenth Annual European Symposium, LNCS 2461, pages 574–585, 2002.
- [KMPS95] A. P. Kamath, R. Motwani, K. Palem, and P. Spirakis. Tail bounds for occupancy and the satisfiability threshold conjecture. *Random Structures and Algorithms*, 7:59–80, 1995.
- [Knu76] D. E. Knuth. *Marriages Stables et leurs relations avec d’autres problèmes combinatoires*. Les Presses de l’Université Montréal, 1976.
- [KS94a] S. Kirkpatrick and B. Selman. Critical behavior in the satisfiability of random Boolean expressions. *Science*, 264(5163):1297–1301, 1994.
- [KS94b] S. Kirkpatrick and B. Selman. Critical behavior in the satisfiability of random formulas. *Science*, 264:1297–1301, 1994.
- [LGM98] M.L. Littman, J. Goldsmith, and M. Mundhenk. The computational complexity of probabilistic planning. *Journal of Automated Reasoning*, 1998.
- [Lit99] M. Littman. Initial experiments in stochastic satisfiability. In *Proc. of the 16th National Conference on Artificial Intelligence*, pages 667–672, 1999.
- [LMP01] M.L. Littman, S.M. Majercik, and T. Pitassi. Stochastic Boolean satisfiability. *Journal of Automated Reasoning*, 27(3):251–296, October 2001.
- [Lov79] L. Lovász. *Combinatorial problems and exercises*. North-Holland, Amsterdam, 1979.
- [Lub66] D. Lubell. A short proof of sperner’s theorem. *J. combinat. Theory*, 1:299, 1966.
- [MdlV95] A. El Maftouhi and W. Fernandez de la Vega. On random 3-sat. *Combinatorics, Probability and Computing*, 4:189–195, 1995.
- [Mes63] L. D. Meschalkin. A generalization of sperner’s theorem on the number of subsets of a finite set. *Theor. Probl, Appl.*, 8:203–204, 1963.
- [MMZ⁺01] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an Efficient SAT Solver. In *Proceedings of the 38th Design Automation Conference (DAC’01)*, 2001.
- [MMZ03] Stephan Mertens, Marc Mézard, and Riccardo Zecchina. Threshold values of random k-sat from the cavity method, 2003.
- [Mou84] Ole G. Mouritsen. *Computer Studies of Phase Transitions and Critical Phenomena*. Springer-Verlag, Berlin Heidelberg, 1984.
- [MSL92] David G. Mitchell, Bart Selman, and Hector Levesque. Hard and easy distributions of SAT problems. *Proc. 10th Nat’l. Conf. on Artificial Intelligence*, pages 459–465, 1992.

- [MU49] N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, pages 335–341, 1949.
- [Orp90] P. Orponen. Dempster’s rule of combination is #P-complete. *Artificial Intelligence*, 44:245–253, 1990.
- [Pap94] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.
- [PY91] C. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Science*, 43:425–440, 1991.
- [Rot96] D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2):273–302, 1996.
- [Sim75] J. Simon. *On some central problems in computational complexity*. PhD thesis, Cornell University, Computer Science Department, 1975.
- [SML96] B. Selman, D. G. Mitchell, and H. J. Levesque. Generating hard satisfiability problems. *Artificial Intelligence*, 81(1-2):17–29, 1996.
- [ST98] John Slaney and Sylvie Thiébaux. On the hardness of decision and optimization problems. *13th European Conference on Artificial Intelligence, ECAI98*, pages 224–248, 1998.
- [Tod89] 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.
- [Val79a] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.
- [Val79b] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.
- [Yam54] K. Yamamoto. Logarithmic order of free distributive lattices. *J. math. Soc. Japan*, 6:343–353, 1954.
- [Zha01] Weixiong Zhang. Phase transitions and backbones of 3-SAT and maximum 3-SAT. *Principles and Practice of Constraint Programming(CP2001), LNCS 2239*, pages 153–167, 2001.