TECHNISCHE
UNIVERSITÄT
WIEN

VIENNA
UNIVERSITY OF
TECHNOLOGY

MASTER THESIS

New Upper Bounds for the SAT/UNSAT Threshold for Shapes

carried out at the

Institute of Information Systems
Knowledge Based Systems Group
of the Vienna University of Technology

under the instruction of

Ao. Univ.-Prof. Dr. Uwe Egly

by

Rocío Santillán Rodríguez

Magdalenenstraße 25/15
1060, Vienna

Vienna, June 2007

....
Rocío Santillán Rodríguez

Words of Thanks

First of all, I thank my parents for its support in my life. I owe them all I am. I thank as well to all my friends and family, who supported me along the completion of the Master program.

To my teachers and professors, both from the program and those involved in my previous education, for their enthusiasm and support. My special thanks to all who supported me with my thesis. I owe special thanks to Nadia Creignou and Hervé Daudé, both teaching in Marsella. One of their papers (accepted, but not published at the time of writing) were crucial for this thesis. Moreover, their explanations on difficult matters were really helpful. They also provided one of the computer programs in `MAPLE` for the experiments. I thank as well to Juan Navarro, whose paper is one of the central ideas in the thesis, for his technical assistance with the LaTeX system, used to write this thesis.

Finally, I thank the European Commission, the "Erasmus Mundus programme" for its financial support. Without the support I simply could not come to study the Master program.

Abstract

Phase transitions not only occur in physics but also in many problems in computer science. Among the problems from computer science, there are many NP-complete ones. An especially important problem is the satisfiability problem, SAT, for Boolean formulas. When phase transitions of SAT problems are studied, the corresponding formulas are usually restricted to conjunctive normal form, i.e., the formula is a set of clauses. K-SAT is the problem, where all clauses consists of exactly K literals. When the density of clauses (i.e., the ratio between the number of clauses and the number of variables) is increased in random K-SAT problems, there is an abrupt change in the probability from being satisfiable to being unsatisfiable. Numerous experimental results are available, but the exact location of the phase transition is not known for the random K-SAT problem with $K > 2$. There are only lower and upper bounds which are rigorously proven.

In this thesis, we consider formulas with more structure, namely the model of fixed balanced shapes introduced by Navarro and Voronkov in 2005. They experimentally studied different shapes and provided first upper bounds for the critical value, i.e., the location where the phase transition occurs. These upper bounds were obtained by using the first moment method (FMM).

We uniformly improve their upper bounds by a method which is based on locally maximal solutions. This method has been proposed by Creignou and Daudé in 2007. Since this method requires a sensitivity polynomial as an input, we show how such polynomials can be computed for shapes, and how the upper bounds are obtained. We discuss the limitations of the method by comparing the upper bounds computed with the FMM with the upper bounds computed by the new method for shapes with increasing size or depth.

Contents

Chapter 1

Introduction

In everyday life, optimization problems appear everywhere. In terms of time and cost, the processes should be minimized, and the gain should be maximized. On many occasions, the aim is to find the best way to execute several tasks. These problems have been abstracted and studied in mathematics and sometimes an elegant algorithm to solve them has been found. Such algorithms are implemented on computers in order to take advantage of the power of speed of a machine. However, there are some problems that, even with the fastest machine, take a long time to be solved.

Problems with different levels of difficulty are separated into classes. One of the most prominent classes is the class NP. This class contains all problems which can be solved by a non-deterministic Turing machine in polynomial time.

It was Stephen Cook in 1971 who introduced the notion of completeness. NP-complete problems are those in NP that are the hardest to solve. Cook showed in [3] that the satisfiability problem SAT and some other problems are NP-complete.

The classification of problems to complexity classes like NP and the proof of completeness always refer to the complexity *in the worst case.* For practical purposes, average case complexity is even more interesting. However, obtaining results for the average case is not an easy task. An important question one has to answer is how the problems are distributed. A pragmatic approach is to consider each problem as equally probable and to investigate randomly generated problem instances.

Pioneering work by Erdös and Rényi [7] describes the behaviour of randomly chosen samples of graphs with respect to some properties. Given a parameter p, $0 \leq p \leq 1$, and a set of n vertices, an edge is added with probability p. Defining $p = c/n$, a structural change on the graph is observed exactly at the point $c = 1$, and a giant component connecting most of the vertices appears. When the process is shown on a chart, one can observe that the change is not smooth but actually has the form of an abrupt *phase transition.*

In the last years, it has been experimentally observed that many problems show a phase transition. Interesting for us here is the 2-SAT problem, i.e., the
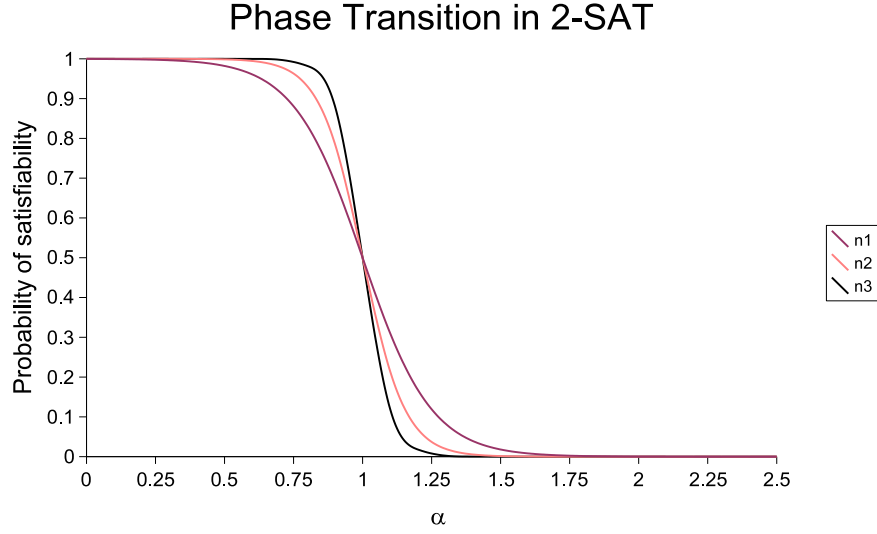
Figure 1.1: The SAT/UNSAT transition for 2-SAT for $n_1 < n_2 < n_3$ variables. Given a number of clauses m, the ratio between clauses and variables, $\alpha = m/n_i$, is the crucial parameter by which we can observe a phase transition in the probability to satisfy a randomly generated formula.

satisfiability problem where any clause has exactly two literals. This problem is in the complexity class P, i.e., it can be solved by a *deterministic* Turing machine in polynomial time. Figure 1.1 shows three curves (for increasing integer numbers n_1, n_2 and n_3) for the random 2-SAT problem, i.e., each instance of the problem is randomly chosen. The curves display, for each value of α, the fraction of satisfiable formulas (or, in other words, the "chance" of a randomly chosen 2-SAT formula to be satisfiable).

The parameter α is the ratio between the number of clauses and the number of variables; it is the crucial parameter for determining the phase transition. In the figure, we see that these phase transitions occur around the critical value $\alpha = 1$. Contrary to many other SAT problems, for which only upper and lower bounds are known, the critical value for α in 2-SAT has been rigorously proven by Chávatal and Reed [2] and independently by Goerdt [12].

The phenomenon where properties show an abrupt change has not only been observed for SAT problems but also for many other ones. The root of the studies of phenomena related to phase transition is physics. In the area of statistical physics, many tools have been developed which are useful in studying "non-physical" phase transitions. Nowadays, the study of threshold phenomena has attracted people from at least three different communities, namely (statistical) physics, mathematics and computer science (see, e.g., [6]).

From a point of view of computer science, not only the phase transition

itself is interesting, but also computational properties of the underlying problem. Since it is extremely hard to obtain rigorous proofs for the critical value where the phase transition occurs, only lower and upper bounds can be provided at the moment. Using implementations to compute randomly chosen instances and analyze the property under consideration (e.g., use a SAT solver on randomly generated clause sets, where each clause has exactly three literals), empirical values can be obtained. The hope is that such empirical values can be used to "guide" the mathematical proofs.

During early experiments with K-SAT, it was observed that there is a relationship between the phase transition and the computational cost to solve a set of randomly chosen problem instances. In the area close to the empirically observed phase transition, the time required to solve the problems strongly increases. The order of the increase depends on the underlying problem; for K-SAT with $K \geq 3$ (and other NP-complete problems), the empirically evaluated increase is exponential. Therefore, the "hard" problems are located around the phase transition. On the other hand, problems located at the beginning and at the end of the curves are easy; thus, an "easy-hard-easy" pattern in the difficulty of the problem has been observed. More details of the phase transition and its relation with the easy-hard-easy pattern will be presented in Chapter 2.

The K-SAT decision problem discussed so far is just a subset of the more general SAT problem. It reduces the nesting of operators to a degree of two and it requires a reduction of the representation of the problem to conjunctive normal form (CNF), with negation occurring only in front of variables. However, many logical representations of real problems result in formulas which have much more structure than just clauses, and a translation step into CNF is required, which is well-known to be problematic.

The aim is to have more general formulas which allow deeper levels of nesting and therefore more structure. A proposal for generalized formulas is presented in [18]. In this paper, Navarro and Voronkov perform some experiments resulting in satisfiability curves (like the ones in Figure 1.1) together with associated computational cost curves. They also observed an easy-hard-easy pattern for the satisfiability problem of shapes.

1.1 Overview and Results

We investigate balanced shapes which can be viewed as a generalization of clauses. For such shapes, Gonzalez and Voronkov provided first upper bounds for the critical value of the phase transition using the first moment method. An important input parameter for this method is the number of satisfying assignments that a randomly chosen shape has.

In order to improve the known upper bounds (i.e., in order to get lower ones), we investigate an approach which is based on *locally maximal solutions* from [4][1].

---

[1]This method was analyzed, for K-SAT only, in [5] under the name of *negatively prime solutions* and it is the best upper bound found for general K-SAT.

The set of all solutions (satisfying assignments) is partitioned following [4] and the sensitivity polynomial is calculated. This calculation is based on the *sensitivity of variables*. In order to avoid complicated computations by hand, we use a program to compute the coefficients of the polynomial according to the recursive formula we develop in Chapter 3. Similar to the number of solutions in the first moment method, the sensitivity polynomial is an input parameter into the new method. The calculation of the new upper bound means then solving a rather complicated equation which is performed with a small program in `MAPLE`. We obtain

- a general method to compute the sensitivity polynomial for balanced shapes,

- a general method (from [4]) to solve the equations, and

- new (and better) upper bounds for all the shapes proposed in [18].

Moreover, we show some absolute and relative improvements of the new method compared to the first moment method.

The structure of the thesis is as follows. In Chapter 2, we introduce some basic definitions and notations. We review some examples (especially random graphs and the travelling salesman problem) for phase transitions from the literature. In Section 2.3, we explain the fundamental ideas underlying the first moment method. The fixed shape model together with balanced shapes are discussed in Section 2.4. Also in Section 2.4, we show how to the first moment method is applied to balanced shapes.

In Chapter 3, we formally introduce the notions of locally maximal solutions and the sensitivity polynomial. We then establish in Section 3.2 the number of solutions for a given shape and continue with the investigation of the sensitivity polynomial in Section 3.3. Theorem 3.4.1 in Section 3.4 summarizes the approach to obtain improved upper bounds for the shapes.

In Chapter 4, we report the results for the shapes discussed in [18] and discuss limitations of the improvement of the chosen method. We present also experimental results with different shapes and remark the presence of some patterns.

In Chapter 5, we conclude with a summary and a discussion about future work.

Chapter 2

Preliminaries

We present definitions and notations which are used later. Next, we discuss phase transitions and review some examples from the literature. Moreover, the first moment method is introduced in detail. Finally, we review the concept of balanced shapes.

2.1 Definitions and Notations

We will follow the standard notation of propositional logic. Let $\mathcal{A}$ be the set of propositional (Boolean) variables. The set of connectives we consider consists of $\neg$ (negation), $\wedge$ (conjunction) and $\vee$ (disjunction). We define the syntax and semantics of propositional logic relative to $\mathcal{A}$.

Definition 2.1.1. The set $\mathcal{L}_\mathcal{A}$ of well-formed Boolean formulas is the smallest set that fulfills the following properties:

1. If A is a variable, $A \in \mathcal{A}$, then $A \in \mathcal{L}_\mathcal{A}$.

2. If $F \in \mathcal{L}_\mathcal{A}$ then $(\neg F) \in \mathcal{L}_\mathcal{A}$.

3. If $F \in \mathcal{L}_\mathcal{A}$ and $G \in \mathcal{L}_\mathcal{A}$ then $(F \wedge G) \in \mathcal{L}_\mathcal{A}$ and $(F \vee G) \in \mathcal{L}_\mathcal{A}$.

In order to save parenthesis, we use the following ranking of binding strength: $\neg$, $\wedge$ and $\vee$. This means that $\neg$ binds stronger than $\wedge$ which in turn binds stronger than $\vee$.

A *literal* is either a variable or the negation of a variable.

A *clause* is a disjunction of literals. A *set* or *collection of formulas* is often associated with the conjunction of its elements.

After having defined the *syntax* of propositional logic, we turn our attention to the *semantics*.

Let 0 and 1 be two *truth values* representing false and true respectively. The semantics of propositional logic is based on *interpretation* functions.

Definition 2.1.2. Let $\mathcal{A}$ be a set of propositional variables. The *interpretation function* for $\mathcal{A}$ is a mapping

$$I : \mathcal{A} \mapsto \{0, 1\} .$$

Interpretation functions are often called variable assignments because variables get a truth value under I. The interpretation function is extended to all elements of $\mathcal{L}_{\mathcal{A}}$ as follows.

$$I(F) = \begin{cases} I(F) & \text{if } F \in \mathcal{A}; \\ 1 - I(G) & \text{if } F \text{ is of the form } \neg G; \\ min(I(G), I(H)) & \text{if } F \text{ is of the form } G \wedge H; \\ max(I(G), I(H)) & \text{if } F \text{ is of the form } G \vee H. \end{cases}$$

Assignments I which make a formula F true are called *models of F*, *satisfying assignments for F* or simply *solutions for F*. If every assignment for F is a solution, then F is a *tautology*. Conversely, if every assignment for F is non-satisfying, then F is called a *contradiction*. If at least one assignment is a solution, then F is *satisfiable*.

Now let us introduce the basic notions of complexity and the SAT problem.

2.1.1 O-notation

The complexity of a problem is a very important concept in theory and in practice. An intuitive definition given in [11] of the time complexity of an algorithm is:

> "The time complexity function for an algorithm expresses its time requirements by giving, for each possible input length, the largest amount of time needed by the algorithm to solve a problem instance of that size."

Definition 2.1.3. [19] Let $f(n)$, $g(n)$ be two non-negative functions defined for all positive integers.

1. It is said that
$$f(n) = O(g(n))$$

   if there exist constants c and n_0 such that, for all $n > n_0$, $f(n) \leq c \cdot g(n)$.

2. It is said that
$$f(n) = \Theta(g(n))$$

   if there exist constants c_1, c_2 and n_0 such that, for all $n > n_0$, $c_1 g(n) \leq f(n) \leq c_2 g(n)$ holds.

O-notation denotes an upper bound that may not be asymptotically tight. Θ-notation is used for denoting upper or lower bounds that are asymptotically tight. For example, if an algorithm needs $3n^2 + 5n$ steps, it runs in time $\Theta(n^2)$.

Some problems encountered in practice are solvable with an algorithm associated with a polynomial function. However, there are also many other problems where the time complexity is an exponential function of the input size. This means that, for some real problems, even a computer can spend too much time looking for the solution, even when the size of the input is not large, say, less than hundreds of thousands of elements. There are also problems for which the order Θ is not known to be polynomial or exponential.

A problem from the complexity class NP (or an NP-complete problem for short) is a problem that can be solved in polynomial time by a *non-deterministic Turing machine (NDTM)*. Informally a NDTM is a machine that "guesses" in each step of the computation between a tree of possible outputs. As a model of computation, it is unrealistic but useful to determine the nature of some difficult problems. For more details about deterministic and non-deterministic TM, consult [11] and [19].

An NP-complete problem is an NP problem whose complexity is such that any other NP-complete problem can be reduced to it in polynomial time.

2.1.2   SAT Problem

SAT is a decision problem associated to the satisfiability of formulas in propositional logic. This problem is well known and, in fact, it was the first problem shown to be NP-complete [3].

The general SAT problem consists of whether, for a given Boolean formula $F(x_1, ..., x_n)$, there is an assignment I of variables x_i to truth values 0 (false) or 1 (true), leading to $I(F(x_1, ..., x_n)) = 1$. This is, F is evaluated to true in propositional logic.

2.1.3 K-SAT and Random K-SAT

Satisfiability problems are often restricted in the sense that only formulas of a specific type are considered. K-SAT (for K an integer greater than 0) is a family of such restricted SAT problems. For the problem K-SAT, the only formulas that are considered, are in conjunctive normal form and the clauses consist of exactly K-literals.

Random K-SAT is the generation of instances of the K-SAT problem in a specific way such that randomness of the instance is guaranteed. In this way, we can analyze properties of the average case rather than the worst case. In general, for random K-SAT, we have a predefined number n of variables, a predefined number K for the size of any clause and a predefined number m of clauses. Therefore, there are $2n$ literals. The K-clauses are generated by choosing each time *randomly and with replacement* a literal, with the restriction that an atom cannot be chosen twice in the same clause. The clauses are, as usual, joined by

conjunction thus having:

$$F = \bigwedge_{i=1}^{m} C_i,$$

where C_i is a clause with exactly K literals. Although this is a standard generation, sometimes it can be different, depending on the properties that are being proved. Hereafter, this will be the method of random generation if we do not specify otherwise.

2.2 Phase Transition

In several combinatorial optimization problems, a phenomenon of structural change in the problem is observed at a certain critical point. In physics, this phenomenon often occurs and has been studied using different approaches. For example, in statistical physics, several analytical and numerical methods have been developed in order to understand the properties of such phenomenon, for example, to analyze the macroscopic thermodynamic behaviour of models.

In a more formal way, one can say that phase transition is "a drastic change in the properties of a system when some external parameter is slightly modified" [13]. This phenomenon has been observed during a long time in the area of statistical mechanics. A common example of a phase transition is the states of matter. For example, in water, at certain critical points for some parameter —in this case the temperature— phase transitions occur; $T = 100°$ C stands for the critical value of the liquid-gas transition and $T = 0°$ C for the critical value of the solid-liquid transition.

2.2.1 Phase Transition by a Structural Parameter

The phase transition phenomenon does not only occur in natural events; it can also be observed when studying more abstract concepts and constructions. We will present here an example in random graph theory.

In this area of mathematics, the behaviour of a typical graph is studied. First, a random graph is generated from a set of vertices. The idea of this ensemble is that an edge is added every time. In this way, a typical graph is studied (instead of a particular instance of a graph) using tools from probability theory. Using this approach, it is easier to see under what conditions a graph with particular properties appears.

The simplest idea [7] consists of a graph $\mathcal{G}(N, p)$ with $|N| = n$ vertices and an associated probability p. Every edge is drawn independently according to this probability. This means that a random number r, $0 \leq r \leq 1$, is assigned to every edge e_r. If $r < p$, then this instance of the graph $\mathcal{G}$ will contain e_r. Thus, for $p = 0$, the graph has no edges and for $p = 1$, the graph is complete (i.e., every vertex is directly connected with all the others).

To imagine the graph construction as an evolutionary process is more intuitive; this process goes on as follows. For all $i, j \in N, i < j$, a random number

$x_{i,j}$ is obtained. The increasing function $p(t)$ represents the probability p at time t, starting from 0 at time $t = 0$. Then t —and therefore $p(t)$— starts to increase. Whenever $p(t)$ exceeds a number $x_{i,j}$, an edge between vertices i and j is added.

It has been proven by Erdős and Rényi [7] that, for a given constant $c > 0$, as long as $c < 1$, the graph $\mathcal{G}(N, c/n)$ contains components of order up to $O(\ln n)$. Then, for $c > 1$, a *giant component* appears, connecting a finite fraction of vertices such that all the other components are still small, with only $O(\ln n)$ vertices.

It is observed that the system undergoes a phase transition at $c = 1$. The number of random graphs where a giant component appears connecting a fraction of all vertices goes from very low to very high at this point. In contrast to physical phenomena, the transition is not induced by an external control parameter like the temperature, but by a structural parameter of the graph. If we make a chart with the percentage of graphs that present a giant component at each point c, we observe a sharp curve at the critical value $c = 1$.

One of the characteristics of phase transition in this example is that the curve becomes sharper when the size of the randomly generated instances is increased. The fraction of vertices that are connected by the giant component grows linearly in $c < 1$ and exponentially in $c > 1$. It approaches the limes of a function that just starts to grow at $c = 1$, as it is explained in [13].

2.2.2 Phase Transition in Combinatorial Optimization Problems

Here we present another example, the travelling salesman problem (TSP) [13], or more specifically, the Euclidean TSP. A special definition based on a random ensemble is used. Let a plane of area size $A = L_x \times L_y$ be given. The random instances consist of n cities, which are randomly placed in the plane. The coordinates $(x_i, y_i) \in [0, L_x] \times [0, L_y]$ denote the position of the city i. The range for x_i is from 0 to L_x and the range for y_i is from 0 to L_y. All positions are equally probable. The distance between pairs of cities is the Euclidean distance, i.e.,

$$d(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$

For each random instance, the following question is asked: *Is the shortest round trip through all cities shorter than a given length l?*. In this way, TSP becomes a decision problem.

The probability p that a tour of length less than l exists is thus presented as a function of the length $\Phi = l/\sqrt{nA}$. Note that the area A is a constant in the whole system. There is a strong increase in p when l is increased close to $\Phi = 0.78$, hence a phase transition occurs. This property is independent from the number of cities, as it can be seen in Figure 2.1.

It has been observed as well that close to the point where half of the instances have a shorter length than l, the running time grows to its maximum. This is, the difficulty of the problem increases in the region close to the phase
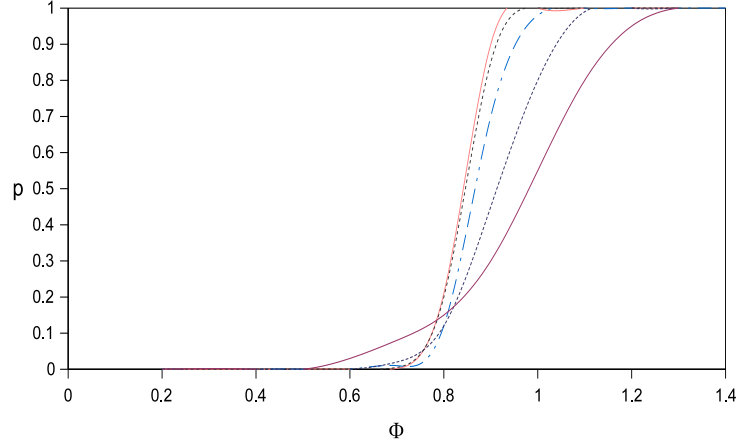
Figure 2.1: Probability p that the minimum tour is shorter than $\Phi = l/\sqrt{nA}$ for n cities. It can be observed that the phase transition becomes sharper when the number of cities is increased. (Figure taken from [13].)

transition. Intuitively, in this specific problem, for the region with small values of l, even the closest cities are a larger distance apart than the given value of l and then the algorithm stops almost immediately. On the other hand, for large l, several permutations of cities have a total distance smaller than l, and then the algorithm stops quickly as well. The region with the phase transition is in the middle of these two. The property to have these three regions is called the *easy-hard-easy pattern*.

2.3   The First Moment Method (FMM)

As mentioned earlier, probability theory is useful in studying the characteristics of the behaviour of the phase transition. When we study a random instance, we have the advantage of knowing that it will present well-studied properties, and the tools of probability theory become available. These tools provide a deeper understanding of the features of the parameters that characterize a phase transition. One of the main tools supplied by probability theory is the *first moment method* (FMM).

The *first moment* or expected value of a random variable can be seen as the average of all the values that the variable can take. It is actually the sum of the probabilities of each possible outcome of the the variable multiplied by the outcomes themselves [17].

Putting it more formally, in the case of a discrete random variable X that

can take values $x_1, ..., x_n$ with the corresponding probabilities $p_1, ..., p_n$, the expected value is

$$E(X) = \sum_i x_i p_i.$$

The core of the first moment method consists of two simple inequalities based on the expected value. The *First Moment Principle* claims the following:

$$\text{If } E(X) \leq t, \text{ then } Pr(X \leq t) > 0, \tag{2.1}$$

with Pr standing for "probability".

Now, how can we justify (2.1). The proof is rather insightful. Since the expected value of X can be seen as an average, if every possible outcome of the random variable X is greater than t, then the average is greater than t. In other words, assuming $Pr(X \leq t) = 0$ holds, then $x_i > t$ holds for all x_i, and $E(X) = \sum_{i, x_i > t} x_i p_i$. This establishes the implication $Pr(X \leq t) = 0 \rightarrow E(X) > t$; therefore, by contraposition, the statement (2.1) follows.

The other important statement for the first moment method, the *Markov's Inequality*, asserts that, for any *non-negative random variable X,*

$$Pr(X \geq t) \leq \frac{E(X)}{t} \tag{2.2}$$

holds. What is the justification for (2.2)? Again, we have $E(X) = \sum_i x_i p_i$. Since X is non-negative, if we remove the values below t, then we obtain the inequality $E(X) \geq \sum_{x_i \geq t} x_i p_i$. Now, the lowest x_i is t and we can say that $\sum_{x_i \geq t} x_i p_i \geq t \sum_{x_i \geq t} p_i$. Observe that $\sum_{x_i \geq t} p_i$ is the same as $Pr(X \geq t)$, so we obtain

$$E(X) \geq t \cdot Pr(X \geq t)$$

from which (2.2) follows immediately.

The first moment method in general consists of judiciously applying these useful inequalities to prove the probability of some property for a particular problem. In order to do so, it is important to make the right choice of the random variable X. It is usually straightforward to find the expected value $E(X)$. So, for X being a non-negative integer value, if $E(X)$ is shown to be less than 1, then, by the inequality (2.1) and the fact that X is a non-negative and discrete variable, $Pr(X = 0)$ is positive. Markov's inequality is also frequently used for X discrete and $E(X) < 1$. If we know that $Pr(X > 0)$, then the inequality $Pr(X \geq 1) \leq E(X)$ becomes significant.

Another property of the expected value is its linearity, i.e.,

$$\mathrm{E}(X_1 + \cdots + X_t) = \mathrm{E}(X_1) + \cdots + \mathrm{E}(X_t)$$

holds.

In fact, in the K-SAT problem, the computation of the expected value becomes a key calculation to improve the upper bound of the location where the phase transition occurs. This will be seen in more detail later with an illustrative example, to which the first moment method is applied.
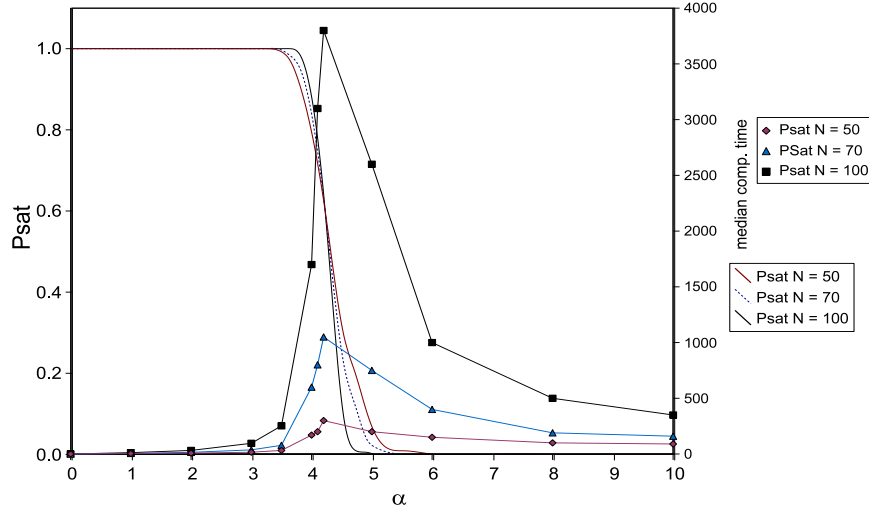
Figure 2.2: The SAT/UNSAT transition for 3-SAT. The relation between the probability of satisfiability Psat and the ratio α is compared to the time required by the DPLL algorithm to solve the randomly generated SAT problems.

According to [13], K-SAT was the first combinatorial decision problem in which experiments showed the presence of a phase transition from satisfiability to unsatisfiability, connected to an easy-hard-easy pattern [16, 14]. The experiments were performed with random instances of the K-SAT problem (we described the random K-SAT in Section 2.1.3). As an example, consider the results shown in Figure 2.2 for 3-SAT and observe the occurring easy-hard-easy pattern.

It has been observed that, using the parameter $\alpha = m/n$, the decrease of probabilities from close to 1 to close to 0 is *not* given by a slowly varying function of α. That is to say that the ratio between the number of clauses and variables is a key value. In the case of $K = 3$, this region of the sharp drop is close to $\alpha \cong 4.26$. The drop becomes increasingly sharper if the number of variables is increased. The instances in the critical region are usually hard to solve, creating an easy-hard-easy pattern. The standard algorithm for SAT is the Davis-Putnam-Logemann-Loveland (DPLL) algorithm which is a branch and bound algorithm that selects a variable and assigns 0 and 1 to it, trying recursively all possible assignments. It uses some heuristics to bound the tree when a formula is known to be unsatisfiable after a partial assignment. A detailed explanation of the DPLL algorithm can be found in [13].

It should be noted that there is only an empirical critical value for K-SAT with $k > 2$; there is no proof, until yet, of where this critical value is exactly located. What is known are upper and lower bounds. The example below consists of the first upper bound obtained for the interval of α where the phase

transition occurs. The example is mostly taken from [15], but this approach is well known and is due to Franco and Paull [9] amongst others. Before we compute the upper bound, we need some preliminaries.

According to [1], the following conjecture, the *Satisfiability Threshold Conjecture*, was formulated first in [2], and has become since then an open problem.

Conjecture 2.3.1. *Let $F_k(n, \alpha n)$ be a random K-SAT formula with n variables and $\alpha n = m$ clauses. Let $S_k(n, \alpha) = Pr(F_k(n, \alpha n)$ is satisfiable). For every $k \geq 2$, there is a constant α_k such that, for all $\epsilon > 0$,*

$$\lim_{n \to \infty} S_k(n, \alpha_k - \epsilon) = 1, \text{ and } \lim_{n \to \infty} S_k(n, \alpha_k + \epsilon) = 0$$

hold.

This conjecture has been stated taking observations from experiments into account. However, there is already a weaker result that was proved by Friedgut [10].

Theorem 2.3.1. *(Friedgut). Let $S_k(n, \alpha) = Pr(F_k(n, \alpha n)$ is satisfiable). For every $k \geq 2$, there is a sequence $\alpha_k(n)$ such that, for any $\epsilon > 0$,*

$$\lim_{n \to \infty} S_k(n, \alpha_k(n) - \epsilon) = 1, \text{ and}$$
$$\lim_{n \to \infty} S_k(n, \alpha_k(n) + \epsilon) = 0$$

hold.

This theorem tells us that at least there exists a threshold for K-SAT such that F_k has a satisfying assignment with probability tending to 0 for $\alpha > 2^k$. By analyzing simple algorithms that obtain satisfying truth assignments with some probability, Franco [8] found a lower bound of α located at $0.9(2^k/k)$; the details of the proof are, however, beyond the scope of this thesis. Nevertheless, the precise point where the phase transition occurs is not known for $k > 2$. The proof that the threshold is a constant has not been found yet. It is possible that the threshold is not a constant, it could even oscillate as a function of n.

Even though it is not yet possible to find a proof of the location of the critical value of α that represents a constant threshold, upper and lower bounds can be obtained.

Let us now consider the example computation of an upper bound for K-SAT. The first proof of a general upper bound uses the first moment method. It was presented first by Franco and Paull in [9].

Let $w_n^1(F)$ be the number of satisfying assignments of a formula F randomly chosen over n variables. There are two approaches here. One of them is to choose a random formula and count the number of variable assignments and the other is to select a random assignment and count the number of formulas it satisfies. Let us here follow the first approach. Using this method, $w_n^1(F)$ can be seen as a random variable, if n, m and K are fixed. From now on, we will write just w_n^1 instead of $w_n^1(F)$ when it refers to the random variable for an arbitrary formula F constructed as explained above.

The expected value $E(w_n^1)$ of the random variable w_n^1 is

$$E(w_n^1) = \sum_F (Pr(F) \cdot w_n^1(F)), \qquad (2.3)$$

where $Pr(F)$ represents the value of the probability that F occurs.

The probability for a random formula of being satisfiable, denoted by $\mathcal{P}_F$, is given by

$$\mathcal{P}_F = \sum_F (Pr(F) \cdot I_F), \qquad (2.4)$$

where

$$I_F = \begin{cases} 1 & \text{if } F \text{ is satisfiable,} \\ 0 & \text{otherwise.} \end{cases}$$

From (2.3), (2.4) and $Pr(w_n^1 \geq 1) = \mathcal{P}_F$, we get a Markov inequality

$$\mathcal{P}_F \leq E(w_n^1) = \overline{w_n^1(F)}. \qquad (2.5)$$

The overbar denotes the average over all the instances of K-SAT.

It is not difficult to evaluate the value of $\overline{w_n^1(F)}$. First we see that each clause forbids 1 of 2^K configurations of the K variables the clause contains. Therefore, the probability that a random assignment I satisfies a clause is $\left(\frac{2^K-1}{2^K}\right)$. Since clauses are drawn independently, the probability to satisfy $m = \alpha n$ clauses is just the multiplication of the probability that I satisfies each clause. Finally, there are 2^n possible assignments so we obtain the formula $2^n \left(\frac{2^K-1}{2^K}\right)^{\alpha n}$ for $\overline{w_n^1(F)}$.

We need to know at what point the random variable $E(w_n^1)$ tends to 0. Since it is a discrete variable, we can just redefine the inequality as $E(w_n^1) < 1$. Hence, we get then the value of α by setting $\overline{w_n^1(F)}$ to 1, i.e.,

$$1 = 2^n \left(\frac{2^K - 1}{2^K}\right)^{\alpha n}.$$

Now we obtain the value for α, i.e.,

$$\alpha = -\frac{\log 2}{\log(1 - 2^{-K})}.$$

This defines a first upper bound for the ratio.

This approach using the FMM was the first successful attempt to get the upper bound of the satisfiability threshold. For $K = 3$, the ratio obtained is $\alpha = -\frac{\log 2}{\log 7 - \log 8} = 5.191$. Refinements reducing the number of satisfying assignments, like considering only maximal solutions, bring results much closer to 4.26. However, the method is the same, so this proof was presented here in order to get a deeper understanding of this method, which will be used later.

2.4 The Fixed Shape Model

The model presented here has been proposed in [18]. Up to now, our formula is in CNF and is represented as a set of clauses. We introduce in the following, more general formulas.

Definition 2.4.1. A *shape* is a propositional formula $\mathcal{S}$ such that

1. $\mathcal{S}$ is built using the conjunction and disjunction connectives only; and

2. every variable appearing in $\mathcal{S}$ has exactly one occurrence in it.

In general, the generation of the random formulas follows the same idea as in K-SAT. There is a predefined shape and an instance of the shape is obtained by choosing atoms at random, together with their sign (which can be positive or negative).

Definition 2.4.2. An *instance* of a shape is any formula obtained by replacing every variable in the shape by a literal from the set of $2n$ literals.

A *randomly generated instance* of a shape $\mathcal{S}$ is a formula obtained by independently and uniformly choosing an element from the set of literals to replace each variable occurring in $\mathcal{S}$.

We introduce *balanced conjunctive shapes* based on balanced trees.

Definition 2.4.3. Given d integers $k_1, ..., k_d$ with $d \geq 0$ and $k_i \geq 2$, the shapes $[\![k_1, ..., k_d]\!]$ and $\langle k_1, ..., k_d \rangle$ are defined recursively as follows.

1. If $d = 0$, then the formulas in both $[\![]\!]$ and $\langle \rangle$ are variables.

2. If $d \geq 1$, then every formula in $[\![k_1, ..., k_d]\!]$ is a conjunction of k_1 formulas in $\langle k_2, ..., k_d \rangle$. Likewise, every formula in $\langle k_1, ..., k_d \rangle$ is a disjunction of k_1 formulas in $[\![k_2, ..., k_d]\!]$.

Later on, shapes of the form $\langle 1, k_2, ..., k_d \rangle$ and $[\![1, k_2, ..., k_d]\!]$ will occur. Such degenerated shapes are unified with shapes $\langle k_2, .., k_d \rangle$ and $[\![k_2, ..., k_d]\!]$, respectively.

We use $li(\mathcal{S})$ to represent the lowest index of a shape $\mathcal{S}$. The value δ, $\delta = d - li(\mathcal{S}) + 1$, represents the *depth* of a shape $\mathcal{S}$.

Observe that K-SAT is a particular case of such shapes, namely the shape $\langle k \rangle$. We will consider instances of shapes as "generalized" clauses and continue to use terms like clause set.

Let us consider an example. The shape $\langle 3, 2 \rangle$ of depth 2 and with shape variables $\{v_1, ..., v_6\}$ can be seen as a tree of the following structure:
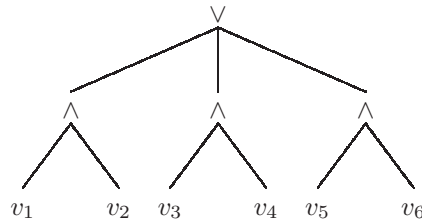
Using the literals of $\{x_1, ..., x_5\}$, examples of instances of this shape are $((x_2 \wedge x_4) \vee (x_1 \wedge \neg x_2) \vee (\neg x_3 \wedge x_5))$ and $((x_3 \wedge x_5) \vee (\neg x_5 \wedge x_1) \vee (\neg x_2 \wedge \neg x_5))$.

We present here detailed versions of the proofs in [18], concerning the behaviour of the shapes in terms of the clauses-to-variables ratio α. They are presented just for the shapes $\langle k_1, ..., k_d \rangle$ because, for conjunctive shapes, the model is the same as $\langle k_2, ..., k_d \rangle$. We use the FMM just as it is described in Section 2.3.

Theorem 2.4.1. *Let t be an arbitrary but fixed truth assignment. The probability $p_{\mathcal{S}}$ that t satisfies a random instance of a shape $\mathcal{S} = \langle k_1, ..., k_d \rangle$, can be calculated as follows.*

$$p_{\mathcal{S}} = \begin{cases} 1/2 \text{ for a shape } \mathcal{S} = \langle \rangle, \\ 1 - (p_{\langle k_2, ... k_d \rangle})^{k_1} \text{ for a shape } \mathcal{S} = \langle k_1, ... k_d \rangle. \end{cases}$$

Proof. In the case of a shape of the form $\langle \rangle$, there are two possible assignments, one of them makes the formula true. Hence $p_{\langle \rangle} = 1/2$.

The second case is proved by a case distinction. Note that the probability is the number of instances of the shape satisfied by a truth assignment, divided by the total number of instances of the shape.

Case $d = 1$. Let $k = k_1$ and note that there are $\binom{n}{k} \cdot 2^k$ instances of each shape.

Fixing an assignment I, we obtain $\binom{n}{k} \cdot 2^k - 1$ instances satisfied by I. For the shape $\langle k \rangle$, the assignment prohibits one of the 2^k configurations, i.e., the one with all literals assigned to false. Therefore, the probability that I makes the instance true is $\frac{2^k - 1}{2^k}$. It should not be surprising that this probability is the same as the one obtained in Section 2.3 since $\langle k \rangle$ represents exactly the K-SAT problem. Then the following chain of equalities hold:

$$p_{\langle k \rangle} = \frac{2^k - 1}{2^k} = 1 - \frac{1}{2^k} = 1 - (1/2)^k = 1 - (p_{\langle \rangle})^k$$

Another way to see $p_{\langle k \rangle} = 1 - (1/2)^k$ is to pick k literals. The probability that the literal is false is $1/2$. The probability that all of them are false is $(1/2)^k$; thus, the probability of at least one of them is true is the complement, namely $1 - (1/2)^k$.

Case $d > 1$. Note that the probability of some shape $[\![k_1, ..., k_d]\!]$ of being unsatisfiable, is the same as the probability $p_{\langle k_1, ..., k_d \rangle}$ of the shape $\langle k_1, ..., k_d \rangle$ of being satisfiable and vice versa.

In a fixed shape $\langle k_1, ..., k_d \rangle$, there are k_1 different instances of $[\![k_2, ..., k_d]\!]$. We need the probability of having all of them unsatisfiable, so we use the property mentioned above and get $(p_{\langle k_2, ... k_d \rangle})^{k_1}$. They are multiplied k_1 times because all of them are drawn independently. Therefore, the probability of at least one of them being unsatisfiable should be $1 - (p_{\langle k_2, ... k_d \rangle})^{k_1}$. $\square$

For shapes, as well as for K-SAT, we know the number n of variables and the number m of clauses of a formula. The clause set is simply a conjunction of instances of the given shape; hence the corresponding decision problem

$\langle k_1, ..., k_d \rangle$-SAT is defined as follows: *Is there a satisfying assignment for a conjunction of given instances of the shape $\langle k_1, ..., k_d \rangle$?* Thus, for $d = 1$, it corresponds exactly to the K-SAT problem.

The *density* $\alpha = m/n$ of the formula is defined as the ratio of the number of clauses divided by the number of variables, just as in K-SAT. Experiments with the shape $\langle 3, 3, 2 \rangle$ in [18] show the presence of a phase transition between satisfiable and unsatisfiable formulas by a change in the parameter α in a small specific region. A first upper bound was obtained for this density for shapes in general.

Theorem 2.4.2. *The probability p that a random instance of $\langle k_1, ..., k_d \rangle$-SAT, with n variables and density α is satisfiable tends to 0 as $n \to \infty$ for all $\alpha > \log 2 / \log (1/p)$ with the value of p calculated as in Theorem 2.4.1.*

Proof. An assignment satisfies a conjunction of $\alpha \cdot n$ instances of $\langle k_1, ..., k_d \rangle$ with probability $p^{\alpha n}$ with p obtained by Theorem 2.4.1. The term $p^{\alpha n}$ is justified because the instances are independently chosen from each other in the random generation procedure. Note that our random variable X is again the number of satisfying assignments.

Since there are 2^n assignments, the expected number of satisfying assignments is $2^n p^{\alpha n}$. This is the expected value $E(X)$ of the random variable X. Since X is discrete, we set the equation to 1 such that we get the value of α for which the random variable tends to 0 by using the *first moment principle* (in an analogous way to the example given in Section 2.3); hence, we obtain $2^n p^{\alpha n} = 1$. Then we apply logarithm to both sides of the equation and get

$$\log 2^n p^{\alpha n} = \log 1$$

which implies $n \log 2 + \alpha n \log p = 0$. Hence,

$$\alpha = -\frac{n \log 2}{n \log p} = -\frac{\log 2}{\log p} = \frac{\log 2}{\log 1 - \log p} = \frac{\log 2}{\log 1/p}$$

as desired. $\qquad \qquad \square$

Chapter 3

Locally Maximal Solutions in the Fixed Shape Model

With the FMM applied to the expected number of solutions, the obtained bound is usually far from the experimental results of the critical point where the phase transition occurs. Better bounds can be obtained by focusing on specific solutions; this results in a reduction of the expected number of solutions. A possible approach developed in [4] is the method of locally maximal solutions. We will analyze with this approach the fixed shape model with the balanced shapes.

3.1   Locally Maximal Solutions

In order to reduce the expected number of solutions $E(S_n)$, we have to find a special characteristic of such solutions. For this, we have to define the concept of a *locally maximal solution*.

Given a truth assignment I for a set of variables $N = \{x_1, ..., x_n\}$, let $\hat{I}_i$ be the truth assignment defined by

$$\hat{I}_i(x_j) = \begin{cases} 1 - I(x_i) & \text{for } j = i \text{ and} \\ I(x_j) & \text{for } j \neq i. \end{cases}$$

This means that we flip the value of the ith variable and let all other variables untouched. Now, given a formula Φ, let I be a satisfying assignment, i.e., $I(\Phi) = 1$. The variable x_i is a *sensitive variable* for this assignment I with respect to Φ, if $\hat{I}_i(\Phi) = 0$, i.e., if we flip the value of this variable, the assignment is not satisfying anymore. Now let $\mathcal{P}_s(I, \Phi)$ be the set of all sensitive variables with respect to I and Φ, i.e.,

$$\mathcal{P}_s(I, \Phi) = \left\{ x_i \mid \hat{I}_i(\Phi) = 0 \right\}.$$

With these notions, we can define locally maximal solutions.

Definition 3.1.1. A solution I of a formula Φ is a *locally maximal solution* of Φ if and only if $\mathcal{P}_s(I, \Phi) \supseteq \{x_i \mid I(x_i) = 0\}$.

In other words, all the variables assigned to 0 are sensitive variables.

Let $w^i(f) = |f^{-1}(i)|$. Since formulas can be defined as boolean functions with k parameters, $f : \{0,1\}^k \to \{0,1\}$, the *weight* of the function, $w^1(f) = |f^{-1}(1)|$, becomes important. Sometimes the parameter f will be omitted from $w^1(f)$ when it is clear from the context. In order to obtain the expected value $E(w^1)$, it is necessary to estimate the number of formulas satisfied by any randomly chosen assignment.

The information given by the weight can be refined to another parameter, the *sensitivity*. This parameter provides information on how the elements of $f^{-1}(1)$ are distributed on the hypercube $\{0,1\}^k$. Let s_f be a sensitivity function over a vector v of inputs for f, defined as

$$s_f(v) = |\{v' \mid f(v') \neq f(v), dist(v, v') = 1\}|,$$

where $dist(v, v')$ denotes the Hamming distance between v and v'. The Hamming distance is the number of elements in the vector v that are different to those of the vector v'. In other words, $s_f(v)$ is the number of assignments which are obtained from v by flipping exactly one variable such that the f-value also flips.

The set of solutions $f^{-1}(1)$ can be partitioned into k sets, each of them with sensitivity j for $0 \leq j \leq k$. Let $\theta_j(f)$ be the cardinality of the subset containing all elements with sensitivity j. We get

$$\theta_j(f) = |\{v \mid f(v) = 1 \text{ and } s_f(v) = j\}|.$$

Using $\theta_j(f)$, we define the *sensitivity polynomial* of a boolean function f as

$$S_f(y) = \sum_{j=0}^{k} \theta_j(f) \cdot y^r.$$

Observe that, since $\sum_{j=0}^{k} \theta_j(f) = |f^{-1}(1)|$, also $S_f(1) = |f^{-1}(1)|$.

The details on how to use the sensitivity polynomial to calculate the expected number of locally maximal solutions of a boolean function can be found in [4]. Now we will go into details about how to obtain the number of satisfying assignments, i.e., $|f^{-1}(1)|$ and the sensitivity polynomial for a given fixed balanced shape.

3.2 Expected Number of Solutions in the Fixed Shape Model

We can compute the number of satisfying assignments for fixed shapes using the inherent symmetry of the operators $\vee$ and $\wedge$. First, we analyze the base cases

and then we generalize them to all shapes. The ideas behind this estimation are the duality of the operations $\wedge$ and $\vee$ in propositional logic and the following observation on the independence of subshapes.

Given a shape $\mathcal{S}_{1,d}^{\wedge} = [\![k_1, ..., k_d]\!]$, it is formed by a k_1-fold conjunction of the shapes $\mathcal{S}_{2,d}^{\vee} = \langle k_2, ..., k_d \rangle$. The number of satisfying assignments, $w^1(\mathcal{S})$, for a shape $\mathcal{S}$ is computed by

$$w^1(\mathcal{S}_{1,d}^{\wedge}) = w^1(\mathcal{S}_{2,d}^{\vee})^{k_1}. \tag{3.1}$$

The reason is that these shapes contain different variables in the leaves since we are dealing with shapes and not with instances of them. Therefore, the $w^1(\mathcal{S}_{2,d}^{\vee})$ satisfying assignments of the first branch $\mathcal{S}_{2,d}^{\vee}$, can be combined with $w^1(\mathcal{S}_{2,d}^{\vee})$ possibilities in each of the remaining $k_1 - 1$ branches, so we have to multiply $w^1(\mathcal{S}_{2,d}^{\vee})$ with itself k_1 times in order to obtain a satisfying assignment of the whole shape $\mathcal{S}_{1,d}^{\wedge}$.

In the same way, given a shape $\mathcal{S}_{1,d}^{\vee} = \langle k_1, ..., k_d \rangle$ with w^0 non-satisfying assignments and $\mathcal{S}_{2,d}^{\wedge} = [\![k_2, ..., k_d]\!]$, the number of non-satisfying assignments of $\mathcal{S}_{1,d}^{\vee}$ is

$$w^0(\mathcal{S}_{1,d}^{\vee}) = w^0(\mathcal{S}_{2,d}^{\wedge})^{k_1}. \tag{3.2}$$

Again we can use the argument that $w^0(\mathcal{S}_{2,d}^{\wedge})$ non-satisfying assignments of the first branch $\mathcal{S}_{2,d}^{\wedge}$ can be combined with $w^0(\mathcal{S}_{2,d}^{\wedge})$ possible non-satisfying assignments in each of the remaining $k_1 - 1$ branches. We multiply $w^0(\mathcal{S}_{2,d}^{\wedge})$ with itself k_1 times in order to obtain a non-satisfying interpretation of the whole shape $\mathcal{S}_{1,d}^{\vee}$.

In the following, we often use the fact that, for any shape $\mathcal{S}$ with v variables, the equation

$$2^v = w^0(\mathcal{S}) + w^1(\mathcal{S}) \tag{3.3}$$

holds. Now we can prove our argument of duality.

Lemma 3.2.1. *Let $\mathcal{S}_{1,d}^{\vee} = \langle k_1, ..., k_d \rangle$ and $\mathcal{S}_{1,d}^{\wedge} = [\![k_1, ..., k_d]\!]$ be two shapes. Let $v = \prod_{i=1}^{d} k_i$ be the number of variables of $\mathcal{S}_{1,d}^{\vee}$ and $\mathcal{S}_{1,d}^{\wedge}$. The weight $w^1(\mathcal{S})$ denotes the number of satisfying assignments of $\mathcal{S}$ and $w^0(\mathcal{S})$ denotes the number of non-satisfying assignments of $\mathcal{S}$. Then, for all $i \in \{0, 1\}$,*

$$w^i(\mathcal{S}_{1,d}^{\vee}) = w^{1-i}(\mathcal{S}_{1,d}^{\wedge}). \tag{3.4}$$

Proof. The proof is by induction on the depth of the formula, $\delta = d - li(\mathcal{S}) + 1$ where $li(\mathcal{S})$ is the lowest index of the shape $\mathcal{S}$. For the special case of $\langle \rangle$ and $[\![ ]\!]$, where $d = 0$, we have $li(\mathcal{S}) = 1$ and $\delta = (d - 1 + 1) = 0$.

Base Case: $\delta = 0$. In this case, the shapes $\mathcal{S}^{\wedge} = [\![ ]\!]$ and $\mathcal{S}^{\vee} = \langle \rangle$ represent only one variable. Hence, there are two possible interpretations, one of them is a solution and the other is not, thus

$$w^i([\![ ]\!]) = 1 = w^{1-i}(\langle \rangle)$$

holds.

Induction Step. Suppose $\delta > 0$ and assume that, for all $\delta' < \delta$, (3.4) hold for $i \in \{0, 1\}$. Recall that a shape of the form $\mathcal{S}_{1,d}^{\vee}$ or $\mathcal{S}_{1,d}^{\wedge}$ with v variables has 2^v possible truth assignments. By the induction hypothesis we have

$$w^i(\mathcal{S}_{2,d}^{\vee}) = w^{1-i}(\mathcal{S}_{2,d}^{\wedge}) \tag{3.5}$$

where $d - 2 + 1 = \delta - 1$.

We perform a case distinction according to the value of i.

Case 1: $i = 1$. Recall that the shape $\mathcal{S}_{1,d}^{\vee}$ is a k_1-fold disjunction of k_1 subshapes of the form $\mathcal{S}_{2,d}^{\wedge}$.

Consider $w^1(\mathcal{S}_{1,d}^{\vee})$. With (3.3) and (3.2), we obtain

$$\begin{aligned} w^1(\mathcal{S}_{1,d}^{\vee}) &= 2^v - w^0(\mathcal{S}_{1,d}^{\vee}) \\ &= 2^v - (w^0(\mathcal{S}_{2,d}^{\wedge}))^{k_1}. \end{aligned}$$

Applying the induction hypothesis with $i = 1$, we get

$$w^1(\mathcal{S}_{1,d}^{\vee}) = 2^v - (w^1(\mathcal{S}_{2,d}^{\vee}))^{k_1}$$

and by (3.1)

$$w^1(\mathcal{S}_{1,d}^{\vee}) = 2^v - w^1(\mathcal{S}_{1,d}^{\wedge}).$$

Therefore, by (3.3),

$$w^1(\mathcal{S}_{1,d}^{\vee}) = w^0(\mathcal{S}_{1,d}^{\wedge}). \tag{3.6}$$

Case 2. $i = 0$. Analogously to Case 1, we just need to apply the equations (3.1), (3.2) and (3.3) and the induction hypothesis appropriately.

Consider $w^0(\mathcal{S}_{1,d}^{\vee})$. With (3.2) and (3.3) we obtain

$$\begin{aligned} w^0(\mathcal{S}_{1,d}^{\vee}) &= (w^0(\mathcal{S}_{2,d}^{\wedge}))^{k_1} \\ &= (2^v - w^1(\mathcal{S}_{2,d}^{\wedge}))^{k_1}. \end{aligned}$$

Applying the induction hypothesis with $i = 0$, we get

$$w^0(\mathcal{S}_{1,d}^{\vee}) = (2^v - w^0(\mathcal{S}_{2,d}^{\vee}))^{k_1}.$$

Finally, with (3.3) and (3.1),

$$\begin{aligned} w^1(\mathcal{S}_{1,d}^{\vee}) &= (w^1(\mathcal{S}_{2,d}^{\vee}))^{k_1} \\ &= w^0(\mathcal{S}_{1,d}^{\wedge}). \end{aligned}$$

This concludes the proof of all cases and thus, of the lemma. $\qquad\square$

Next, we compute $w^1(\mathcal{S})$ for a shape $\mathcal{S}$.

Lemma 3.2.2. *Let $\mathcal{S}_{1,d}^{\vee} = \langle k_1, ..., k_d \rangle$ be a shape of depth δ with $v = \prod_{i=1}^{d} k_i$ variables. Then the weight of $\mathcal{S}_{1,d}^{\vee}$ is given by*

$$w^1(\mathcal{S}_{1,d}^{\vee}) = \begin{cases} 1 & \text{if } \mathcal{S}_{1,d}^{\vee} = \langle \rangle, \text{ i.e., } \delta = 0; \\ 2^k - w^1(\langle k_2, ..., k_d \rangle)^{k_1} & \text{if } \delta > 0. \end{cases}$$

Proof. For $\mathcal{S}_{1,d}^{\vee} = \langle \rangle$ (i.e., $\delta = 0$), we have $w^1(\mathcal{S}_{1,d}^{\vee}) = 1$.
    For $\delta > 0$, let $\mathcal{S}_{1,d}^{\vee} = \langle k_1, ..., k_d \rangle$. By (3.3) we have

$$w^1(\mathcal{S}_{1,d}^{\vee}) = 2^v - w^0(\langle k_1, ..., k_d \rangle).$$

By applying Lemma 3.2.1, we get

$$w^1(\mathcal{S}_{1,d}^{\vee}) = 2^v - w^1(\llbracket k_1, ..., k_d \rrbracket).$$

Finally, with (3.1), we get

$$w^1(\mathcal{S}_{1,d}^{\vee}) = 2^v - w^1(\langle k_2, ..., k_d \rangle)^{k_1}$$

as desired. $\square$

Next we establish the sensitivity polynomial on which the calculation of the new upper bound is based.

3.3 Sensitivity Polynomial for the Fixed Shape Model

So far, we can compute the number of solutions, the weight, for a shape. It is important to refine this information in order to obtain the distribution of the solutions in $\{0,1\}^k$, as it has been mentioned in Section 3.1.
    Before considering the computation of the sensitivity polynomial for the shapes in a general form, we analyze some simple shapes in order to illustrate how to compute the sensitivity polynomial.

Example 3.3.1. Let $\mathcal{S} = \langle 3 \rangle$, i.e., the shape represents clauses with exactly three literals. There are $2^3 - 1 = 7$ satisfying assignments given below ($\sharp 1$ is the number of 1's occurring in the assignment). We perform for each assignment an analysis of the sensitivity.

No	x_1	x_2	x_3	$\sharp 1$	Sensitivity
1	0	0	1	1	1-sensitive
2	0	1	0	1	1-sensitive
3	1	0	0	1	1-sensitive
4	1	1	0	2	0-sensitive
5	1	0	1	2	0-sensitive
6	0	1	1	2	0-sensitive
7	1	1	1	3	0-sensitive

25

Consider the first assignment. When we flip the assignment for x_3 to 0, then the resulting assignment is no longer satisfying. Since this 1 is the only one in the assignment, this first interpretation is 1-sensitive. This is true, by symmetry, for all assignments with $\sharp 1 = 1$, i.e., for the second and the third. For all the other assignments, flipping a bit has no impact on the valuation of the formula; the resulting assignments remain satisfying. Hence, all the $2^3 - 1 - 3 = 4$ assignments are 0-sensitive.

Summarizing the discussion, we obtain

$$S_{\langle 3 \rangle}(y) = 3y^1 + 4y^0$$

as a sensitivity polynomial for $\mathcal{S} = \langle 3 \rangle$.

The disjunction behaves similarly when k is increased. Next, we analyze the behaviour of the conjunction.

Example 3.3.2. Consider $\mathcal{S} = [\![3]\!]$, i.e., a conjunction of three variables. Then, only the assignment which sets all variables to 1 is satisfying. If we flip any of these occurrences of 1, the resulting assignment is no longer satisfying. Hence, this assignment is 3-sensitive and the sensitivity polynomial is

$$S_{[\![3]\!]}(y) = y^3.$$

With the following examples, the behaviour of shapes with more than one level of nesting is illustrated.

Example 3.3.3. Consider the shape $\mathcal{S} = \langle 2, 2 \rangle$, i.e., the shape has the form $(x_1 \wedge x_2) \vee (x_3 \wedge x_4)$. This shape has four variables $x_1, ..., x_4$ and 2^4 assignments. With Lemma 3.2.2, we compute $w^1(\mathcal{S})$ recursively as follows:

$$
\begin{aligned}
w^1(\mathcal{S}) &= 2^4 - w^1(\langle 2 \rangle)^2 \\
&= 2^4 - (2^2 - 1)^2 \\
&= 16 - 9 \\
w^1(\mathcal{S}) &= 7.
\end{aligned}
$$

We summarize the investigation of the sensitivity for each of the seven assignments in the following table.

No	x_1	x_2	x_3	x_4	Sensitivity
1	1	1	0	0	2-sensitive
2	1	1	0	1	2-sensitive
3	1	1	1	0	2-sensitive
4	1	1	1	1	0-sensitive
5	0	0	1	1	2-sensitive
6	0	1	1	1	2-sensitive
7	1	0	1	1	2-sensitive

Hence, the sensitivity polynomial for $\mathcal{S}$ is

$$S_{\langle 2,2 \rangle}(y) = 6y^2 + 1y^0.$$

Observe that we have a twofold disjunction, for which we examine all combinations of assignments for the immediate subshapes. In the case here, the subshape is $[\![2]\!]$. This subshape has one satisfying assignment and three non-satisfying assignments. We combine all these assignments in order to obtain satisfying assignments for $\langle 2, 2 \rangle$. Whenever we have exactly at one position of the disjunction a satisfying assignment (and at all other positions non-satisfying ones), then the sensitivity of the satisfying assignment of $[\![2]\!]$ is "inherited" to the sensitivity of $\langle 2, 2 \rangle$. Since we have one satisfying and three non-satisfying assignments, which can occur at $k_2 = 2$ places, we have 6 assignments which are 2-sensitive. The remaining satisfying assignment is 0-sensitive.

Example 3.3.4. As a final example, we consider the shape $\mathcal{S} = [\![3, 2]\!]$. This shape has 6 variables $x_1, ..., x_6$ and 2^6 assignments. With (3.3), Lemma 3.2.1 and Lemma 3.2.2, we compute $w^1(\mathcal{S})$ as follows:

$$\begin{aligned}
w^1([\![3, 2]\!]) &= w^0(\langle 3, 2 \rangle) \\
&= 2^6 - w^1(\langle 3, 2 \rangle) \\
&= 2^6 - (2^6 - (w^1(\langle 2 \rangle))^3) \\
&= 2^6 - (2^6 - (2^2 - (w^1(\langle \rangle))^2)^3) \\
&= 2^6 - 2^6 + (4 - 1)^3 \\
w^1([\![3, 2]\!]) &= 27
\end{aligned}$$

Observe that the shape $\mathcal{S}$ is a conjunction of three shapes $\langle 2 \rangle$. We have to consider the satisfying assignments of $\langle 2 \rangle$, which are

$$A = \{(1, 0), (0, 1), (1, 1)\}.$$

Although all these assignments are satisfying, they behave quite differently with respect to sensitivity. We partition A into two disjoint subsets, namely

$$A_1 = \{(0, 1), (1, 0)\}$$

and

$$A_{11} = \{(1, 1)\}.$$

The elements of A_1 are 1-sensitive, (1,1) is 0-sensitive. We have to consider all possibilities for the three conjuncts from A_1 and A_{11}. We have

1. 1 possibility where all elements come from A_{11}.

2. $3 \cdot 2 = 6$ possibilities where exactly one element comes from A_1.

3. $3 \cdot 1 \cdot 2 \cdot 2 = 12$ possibilities where exactly two elements come from A_1.

4. $2 \cdot 2 \cdot 2 = 8$ possibilities where three elements come from A_1.

The only element of case 1. is 0-sensitive, the 6 possibilities of case 2. are 1-sensitive, the 12 possibilities of case 3. are 2-sensitive and the remaining 8 possibilities are 3-sensitive. Therefore, the sensitivity polynomial is

$$S_{[\![3,2]\!]}(y) = 8y^3 + 12y^2 + 6y + 1.$$

The following lemma generalizes the simplest cases (Examples 3.3.1 and 3.3.2) to shapes with arbitrary k and with depth less than 2.

Lemma 3.3.1. *Given a shape $\mathcal{S}$ with depth less than 2, the sensitivity polynomial for $\mathcal{S}$ is the following:*

$$S_{\mathcal{S}}(y) = \begin{cases} y & \text{if } \mathcal{S} = \langle\rangle \text{ or } \mathcal{S} = [\![\,]\!], \\ ky + (2^k - k - 1) & \text{if } \mathcal{S} = \langle k \rangle, \\ y^k & \text{if } \mathcal{S} = [\![k]\!]. \end{cases}$$

Proof. The first case is very simple. There is only one satisfying assignment for the variable, and, by flipping its value, it becomes unsatisfiable. There are no more variables, therefore $\theta_1(\mathcal{S}) = 1$ and this implies $S_{\mathcal{S}}(y) = 1 \cdot y^1 = y$.

The second case is the disjunction. Observe that a satisfying assignment with two or more variables assigned to 1 has no neighbors $(x_1, ..., x_k)$ within distance 1 on which $f(x_1, ..., x_k) = 0$. There are 2^k assignments, from which one is non-satisfying and k have only one variable assigned to 1. Therefore, there are $2^k - k - 1$ satisfying assignments that are 0-sensitive. The other k satisfying assignments are within distance 1 from $\{0\}^k$, so all of them are 1-sensitive, and we get $S_{\mathcal{S}}(y) = ky^1 + (2^k - k - 1)y^0 = ky + (2^k - k - 1)$.

The third case of a conjunction follows a similar analysis. There is only one satisfying assignment, and it has k neighbors within distance 1 that are already non-satisfying. Therefore, it is k-sensitive and $S_{\mathcal{S}}(y) = 1 \cdot y^k = y^k$. $\qquad\square$

From the examples and Lemma 3.3.1, we can generalize to any shape. The description of the sensitivity as a polynomial will prove useful. For the case of a conjunction, let us recall the definition of the multiplication of two polynomials which will be needed later on.

Definition 3.3.1. Given two polynomials

$$\begin{aligned} P(y) &= p_0 y^0 + \cdots + p_n y^n \text{ and} \\ Q(y) &= q_0 y^0 + \cdots + q_m y^m, \end{aligned}$$

the multiplication of them is given by

$$P(y) \cdot Q(y) = \sum_{i=0}^{m+n} \left(\sum_{j=0}^{i} p_j \cdot q_{i-j} \right) \cdot y^i.$$

The multiplication is naturally extended to the exponential operation in the usual way, for example $(P(y))^2 = P(y) \cdot P(y)$.

We will also use the well known Binomial Theorem of combinatorics.

Theorem 3.3.1. *Given a non-negative integer n, the following holds.*

$$(x + y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k.$$

Now we can proceed with the computation of $S(y)$ for a general shape. The main idea is the analysis of the behaviour of $\wedge$ and $\vee$. In the case of $\wedge$ in the top-most level, one has to analyze only the case where all the lower levels are satisfied. Somehow, it "inherits" the sensitivities from the lower levels like in Example 3.3.3. In the case of $\vee$, it is necessary to take into account two cases: either there is one satisfiable branch and all the others are unsatisfiable or there is more than one satisfiable branch. In the first case, the sensitivity depends on lower levels, and in the second, there is no sensitive variable at all. We need to use some combinatorics in order to obtain the number of assignments for each case. From now on, let us call the conjunctive shapes $\mathcal{S}_{i,j}^{\wedge} = [\![k_i, k_{i+1}, ..., k_j]\!]$ and the disjunctive shapes $\mathcal{S}_{i,j}^{\vee} = \langle k_i, k_{i+1}, ..., k_j \rangle$.

Lemma 3.3.2. *Given a shape $\mathcal{S}$ with v variables and depth $\delta > 0$, the sensitivity polynomial $S(y)$ is computed recursively as follows:*

$$S_{\mathcal{S}}(y) = \begin{cases} (S_{\mathcal{S}_{2,d}^{\vee}}(y))^{k_1} & \text{if } \mathcal{S} = \mathcal{S}_{1,d}^{\wedge}, \\ k_1 w^0 (\mathcal{S}_{2,d}^{\wedge})^{k_1 - 1} \cdot S_{\mathcal{S}_{2,d}^{\wedge}}(y) + \\ \sum_{j=2}^{k_1} \binom{k_1}{j} \cdot w^1 (\mathcal{S}_{2,d}^{\wedge})^j \cdot w^0 (\mathcal{S}_{2,d}^{\wedge})^{k_1 - j} & \text{if } \mathcal{S} = \mathcal{S}_{1,d}^{\vee}. \end{cases}$$

Proof. Recall that $\delta = d - li(\mathcal{S}) + 1$. The proof is by induction on the depth δ of the shape.

Base Case: $\delta = 1$. By Lemma 3.3.1, we can prove the base cases as follows. For $\mathcal{S} = [\![k_1]\!]$, we have $S_{\mathcal{S}} = y^{k_1}$ and for $\mathcal{S} = \langle k_1 \rangle$, we have:

$$S_{\langle k_1 \rangle}(y) = k_1 w^0 ([\![]\!])^{k_1 - 1} \cdot S_{[\![]\!]}(y) + \sum_{j=2}^{k_1} \binom{k_1}{j} \cdot w^1 ([\![]\!])^j \cdot w^0 ([\![]\!])^{k_1 - j}$$

$$= k_1 \cdot 1^{k_1 - 1} \cdot y + \sum_{j=2}^{k_1} \binom{k_1}{j} \cdot 1^j \cdot 1^{k_1 - j}$$

$$= k_1 y + \sum_{j=2}^{k_1} \binom{k_1}{j}$$

$$= k_1 y + \sum_{j=0}^{k_1} \binom{k_1}{j} - \binom{k_1}{1} - \binom{k_1}{0}$$

$$S_{\langle k_1 \rangle}(y) = k_1 y + (2^{k_1} - k_1 - 1).$$

Induction Step. Let $\delta > 1$ and assume that, for all $\delta' < \delta$, the lemma holds.

Consider a shape of depth δ. We perform a case distinction according to the type of the shape.

Case 1. Consider $\mathcal{S}_{1,d}^{\wedge}$. We want to show that

$$S_{\mathcal{S}_{1,d}^{\wedge}}(y) = (S_{\mathcal{S}_{2,d}^{\vee}}(y))^{k_1}$$

holds. We perform an inner induction proof on k_1.

Base Case (for Case 1): $k_1 = 1$. Let the shape $\mathcal{S}_{1,d}^{\wedge}$ be $[\![1, ..., k_d]\!]$ which is, by definition, equivalent to $\mathcal{S}_{2,d}^{\vee} = \langle k_2, ..., k_d \rangle$. Therefore, we immediately obtain

$$S_{\mathcal{S}_{1,d}^{\wedge}}(y) = S_{\mathcal{S}_{2,d}^{\vee}}y = (S_{\mathcal{S}_{2,d}^{\vee}}(y))^1.$$

Induction Step (for Case 1). Let $k_1 > 1$ and assume that

$$S_{[\![k_1-1, ..., k_d]\!]}(y) = S_{\mathcal{S}_{2,d}^{\vee}}(y)^{k_1-1}.$$

Consider the shape $\mathcal{S}_{1,d}^{\wedge} = [\![k_1, ..., k_d]\!]$ with $v = \prod_{i=1}^{d} k_i$ variables. Let v_U be the number of variables for $\mathcal{S}_{2,d}^{\vee}$. Let J be a partial assignment for the first $v_T = v - v_U$ variables and let J' be the partial assignment for the last v_U variables. Note that $v_T = (k_1 - 1) \cdot v_U$. Let $I(x) = J(x)$, if x is one of the first v_T variables and let $I(x) = J'(x)$ if x is one of the last v_U variables. $\mathcal{S}_{1,d}^{\wedge}$ can be seen as the formula $[\![k_1 - 1, ..., k_d]\!] \wedge \mathcal{S}_{2,d}^{\vee}$. Let us call

$$T(y) = S_{[\![k_1-1, ..., k_d]\!]}(y) = t_0 y_0 + \cdots + t_{v_T} y^{v_T}$$

and

$$U(y) = S_{\mathcal{S}_{2,d}^{\vee}}(y) = u_0 y_0 + \cdots + u_{v_U} y^{v_U}.$$

For an arbitrary but fixed I, we have two cases.

Subcase 1.1. The assignments J or J' are non-satisfying. In this case, I is non-satisfying and is not taken into account for the sensitivity polynomial for $\mathcal{S}_{1,d}^{\wedge}$.

Subcase 1.2. Both J and J' are satisfying. In this case, let J be j-sensitive and J' be j'-sensitive. Obviously, our interpretation I will be i-sensitive, for $i = j + j'$. Moreover, note that there are t_j j-sensitive interpretations and $u_{j'}$ j'-sensitive interpretations defined by the sensitivity polynomials $T(y)$ and $U(y)$. Note that, for a fixed J, there are then $u_{j'}$ possible assignments to extend it to some I, and conversely, there are t_j possible different extensions for a fixed J'. Therefore, we get $t_j \cdot u_{j'}$ i-sensitive interpretations.

In order to count how many interpretations of this type exist, note that we can express i as $i + 1$ different sums of the form

$$\sum_{j=0}^{i} j + (i - j).$$

We can express the number of i-sensitive interpretations as the sum

30

$$\sum_{j=0}^{i} t_j \cdot u_{i-j}.$$

This holds for every i from 0 to $v = v_T + v_U$ because the sensitivity is at most v. Therefore, we get

$$\sum_{i=0}^{v} (\sum_{j=0}^{i} t_j \cdot u_{i-j}) y^i,$$

which is nothing but $T(y) \cdot U(y)$. Now, since by the induction hypothesis for the inner induction, $T(y) = (S_{\mathcal{S}_{2,d}^{\vee}}(y))^{k_1-1}$, and by definition $U(y) = S_{\mathcal{S}_{2,d}^{\vee}}(y)$, it follows that $T(y) \cdot U(y) = S_{\mathcal{S}_{2,d}^{\vee}}(y)^{k_1}$.

This concludes the inner induction and therefore the proof for Case 1.

Case 2. Consider $\mathcal{S}_{1,d}^{\vee}$. In order to prove

$$S_{\mathcal{S}_{1,d}^{\vee}}(y) = k_1 w^0 (\mathcal{S}_{2,d}^{\wedge})^{k_1-1} \cdot S_{\mathcal{S}_{2,d}^{\wedge}}(y) + \sum_{j=2}^{k_1} \binom{k_1}{j} \cdot w^1 (\mathcal{S}_{2,d}^{\wedge})^j \cdot w^0 (\mathcal{S}_{2,d}^{\wedge})^{k_1-j},$$

we perform an inner induction on k_1 like in the previous case.

Base Case (for Case 2). $k_1 = 1$. Let the shape $\mathcal{S}_{1,d}^{\vee}$ be $\langle 1, ..., k_d \rangle$. We have only one branch in the top-most disjunction implying that this shape is equivalent to $\mathcal{S}_{2,d}^{\wedge} = [\![k_2, ..., k_d]\!]$ by definition. Also note that the second term of the formula $S_{\mathcal{S}_{1,d}^{\vee}}(y)$ is equal to 0 because $k_1 < j$. We have

$$\begin{aligned} S_{\mathcal{S}_{1,d}^{\vee}}(y) &= k_1 w^0 ([\![k_2, ..., k_d]\!])^{k_1-1} \cdot S_{[\![k_2,...,k_d]\!]}(y) + 0 \\ &= 1 \cdot (w^0 ([\![k_2, ..., k_d]\!]))^0 \cdot S_{[\![k_2,...,k_d]\!]}(y) \\ &= S_{[\![k_2,...,k_d]\!]}(y). \end{aligned}$$

Induction Step (for Case 2). Let $k_1 > 1$ and assume that

$$\begin{aligned} S_{[\![k_1-1,...,k_d]\!]}(y) = &(k_1 - 1) w^0 (\mathcal{S}_{2,d}^{\wedge})^{k_1-2} \cdot S_{\mathcal{S}_{2,d}^{\wedge}}(y) \\ &+ \sum_{j=2}^{k_1-1} \binom{k_1-1}{j} \cdot w^1 (\mathcal{S}_{2,d}^{\wedge})^j \cdot w^0 (\mathcal{S}_{2,d}^{\wedge})^{k_1-1-j}. \end{aligned} \tag{3.7}$$

Consider the shape $\mathcal{S}_{1,d}^{\vee} = \langle k_1, ..., k_d \rangle$ with $v = \prod_{i=1}^{d} k_i$ variables. Let v_U be the number of variables for $\mathcal{S}_{2,d}^{\wedge}$. Let J be a partial assignment for the first $v_T = v - v_U$ variables and let J' be a partial assignment for the last v_U variables. We have $I(x) = J(x)$, if x is one of the first v_T variables, and $I(x) = J'(x)$ if x is one of the last v_U variables. The shape we analyze is $\langle k_1 - 1, ..., k_d \rangle \vee \mathcal{S}_{2,d}^{\wedge}$. Let

$$T(y) = S_{\langle k_1-1,...,k_d \rangle}(y) = t_0 y_0 + \cdots + t_{v_T} y^{v_T}$$

and

$$U(y) = S_{\mathcal{S}_{2,d}^{\wedge}}(y) = u_0 y_0 + \cdots + u_{v_U} y^{v_U}.$$

31

We have to consider four subcases for a fixed I. All of them except Subcase 2.1 contribute to the sensitivity polynomial.

Subcase 2.1. J and J' are non-satisfying. In this case, we do not add anything to the polynomial $S_{\mathcal{S}_{1,d}^{\vee}}$.

Subcase 2.2. J is satisfying and J' is non-satisfying.

In this case, the sensitivity of I is the sensitivity of J, i.e., $T(y)$ because J' is non-satisfying; thus, J' does not contribute to the sensitivity. Therefore, for this case, the number of i-sensitive interpretations is exactly t_i, so we just have to multiply them by the number of possible non-satisfying assignments J'.

From now on, we call $w_T^1 = w^1(\langle k_1 - 1, ..., k_d \rangle)$ and $w_U^1 = w^1(\mathcal{S}_{2,d}^{\wedge})$. Conversely, let $w_T^0 = w^0(\langle k_1 - 1, ..., k_d \rangle)$ and $w_U^0 = w^0(\mathcal{S}_{2,d}^{\wedge})$. By the induction hypothesis of the inner induction (3.7), the sensitivity polynomial for this case is

$$w_U^0 \cdot T(y) = w_U^0 \Bigg((k_1 - 1)(w_U^0)^{k_1 - 2} \cdot U(y)$$
$$+ \sum_{j=2}^{k_1 - 1} \binom{k_1 - 1}{j} \cdot (w_U^1)^j \cdot (w_U^0)^{k_1 - 1 - j} \Bigg).$$

Now we simplify this equation so that it can be used later on.

$$w_U^0 \cdot T(y) = (k_1 - 1)(w_U^0)^{k_1 - 1} \cdot U(y) + w_U^0 \sum_{j=2}^{k_1 - 1} \binom{k_1 - 1}{j}(w_U^1)^j (w_U^0)^{k_1 - 1 - j}$$

$$= (k_1 - 1)(w_U^0)^{k_1 - 1} \cdot U(y) + \sum_{j=2}^{k_1 - 1} \binom{k_1 - 1}{j}(w_U^1)^j (w_U^0)^{k_1 - j}$$

$$= (k_1 - 1)(w_U^0)^{k_1 - 1} \cdot U(y) + \sum_{j=0}^{k_1 - 1} \binom{k_1 - 1}{j}(w_U^1)^j (w_U^0)^{k_1 - j} -$$

$$\binom{k_1 - 1}{1}(w_U^1)^1 (w_U^0)^{k_1 - 1} - \binom{k_1 - 1}{0}(w_U^1)^0 (w_U^0)^{k_1 - 0}$$

$$= (k_1 - 1)(w_U^0)^{k_1 - 1} \cdot U(y) + \sum_{j=0}^{k_1 - 1} \binom{k_1 - 1}{j}(w_U^1)^j (w_U^0)^{k_1 - j} -$$

$$(k_1 - 1)w_U^1 (w_U^0)^{k_1 - 1} - (w_U^0)^{k_1}$$

$$= (k_1 - 1)(w_U^0)^{k_1 - 1} \cdot U(y) + w_U^0 \sum_{j=0}^{k_1 - 1} \binom{k_1 - 1}{j}(w_U^1)^j (w_U^0)^{k_1 - 1 - j} -$$

$$(k_1 - 1)w_U^1 (w_U^0)^{k_1 - 1} - (w_U^0)^{k_1}.$$

By the Binomial Theorem 3.3.1, we have

$$w_U^0 \cdot T(y) = (k_1 - 1)(w_U^0)^{k_1 - 1} \cdot U(y) + w_U^0 (w_U^1 + w_U^0)^{k_1 - 1} -$$
$$(k_1 - 1)w_U^1 (w_U^0)^{k_1 - 1} - (w_U^0)^{k_1}.$$

Subcase 2.3. J is non-satisfying and J′ is satisfying. In this case, the sensitivity of I is given by $J′$. We use (3.2) to obtain the sensitivity polynomial in this case, i.e.,

$$w_T^0 \cdot U(y) = (w_U^0)^{k_1-1} U(y).$$

Subcase 2.4. J and J′ are satisfying. In this case, since we have a disjunctive shape, we find that the sensitivity of I is 0, because if we flip any of its variables, the result of the evaluation will not be affected.

Recall that $w_T^1 = 2^{v_T} - w_T^0$. By (3.2), $w_T^1 = 2^{v_T} - (w_U^0)^{k_1-1}$. Thus, the number of possibilities of this type are

$$
\begin{aligned}
w_T^1 \cdot w_U^1 &= (2^{v_T} - (w_U^0)^{k_1-1})w_U^1 \\
&= (2^{v_U(k_1-1)} - (w_U^0)^{k_1-1})w_U^1 \\
&= ((2^{v_U})^{k_1-1} - (w_U^0)^{k_1-1})w_U^1 \\
&= ((w_U^1 + w_U^0)^{k_1-1} - (w_U^0)^{k_1-1})w_U^1 \\
&= w_U^1(w_U^1 + w_U^0)^{k_1-1} - w_U^1(w_U^0)^{k_1-1}.
\end{aligned}
$$

Now we sum up all the different cases in order to obtain the new polynomial:

$$
\begin{aligned}
S(y) = \underbrace{(k_1 - 1)(w_U^0)^{k_1-1} \cdot U(y)}_{A_1} + w_U^0(w_U^1 + w_U^0)^{k_1-1} - \\
(k_1 - 1)w_U^1(w_U^0)^{k_1-1} - (w_U^0)^{k_1} + \\
\underbrace{(w_U^0)^{k_1-1}U(y)}_{A_2} + \\
w_U^1(w_U^1 + w_U^0)^{k_1-1} - w_U^1(w_U^0)^{k_1-1}.
\end{aligned}
$$

Observing that the final equation we want to obtain is of the form $A \cdot U(y) + B$, we marked terms that will correspond to part $A \cdot U(y)$ and the rest forms part B. Now we sum up $A_1 + A_2$:

$$A \cdot U(y) = (k_1 - 1)(w_U^0)^{k_1-1}U(y) + (w_U^0)^{k_1-1}U(y) = k_1(w_U^0)^{k_1-1}U(y)$$

The rest forms part B of our goal expression:

$$
\begin{aligned}
B = w_U^0(w_U^1 + w_U^0)^{k_1-1} - (k_1 - 1)w_U^1(w_U^0)^{k_1-1} - (w_U^0)^{k_1} + \\
w_U^1(w_U^1 + w_U^0)^{k_1-1} - w_U^1(w_U^0)^{k_1-1}.
\end{aligned}
$$

By factorizing $(w_U^1 + w_U^0)^{k_1-1}$ from the first and fourth term, we get

$$
\begin{aligned}
B &= (w_U^1 + w_U^0)^{k_1-1}(w_U^0 + w_U^1) - (k_1 - 1)w_U^1(w_U^0)^{k_1-1} - \\
&\quad (w_U^0)^{k_1} - w_U^1(w_U^0)^{k_1-1} \\
&= (w_U^1 + w_U^0)^{k_1} - k_1 w_U^1(w_U^0)^{k_1-1} - (w_U^0)^{k_1}.
\end{aligned}
$$

Applying the Binomial Theorem 3.3.1 yields

$$
\begin{aligned}
B &= \left(\sum_{j=0}^{k_1} \binom{k_1}{j} (w_U^1)^j \cdot (w_U^0)^{k_1-j} \right) - k_1 w_U^1 (w_U^0)^{k_1-1} - (w_U^0)^{k_1} \\
&= \left(\sum_{j=0}^{k_1} \binom{k_1}{j} (w_U^1)^j \cdot (w_U^0)^{k_1-j} \right) - \binom{k_1}{1} (w_U)^1 (w_U^0)^{k_1-1} - \\
&\quad \binom{k_1}{0} (w_U^1)^0 (w_U^0)^{k_1-0} \\
&= \sum_{j=2}^{k_1} \binom{k_1}{j} (w_U^1)^j \cdot (w_U^0)^{k_1-j}
\end{aligned}
$$

Now we just sum up $A \cdot U(y)$ and B and we get exactly $S(y)$.

This concludes the inner induction for the second case, and thus, the proof of the lemma $\qquad\square$

3.4 A Better Upper Bound of K-SAT

The basic idea in the calculation of the new upper bound is that, instead of focusing on the whole set of solutions expressed as $E_{n,cn}(\texttt{Sol})$, it applies the FMM to the set of locally maximal solutions $E_{n,cn}(\texttt{MaxSol})$.

The following theorem, shows how the upper bound is computed when the sensitivity polynomial $S(y)$ is known. $S'(y)$ is used to denote the (first) derivative of a polynomial $S(y)$.

Theorem 3.4.1. *[4, Theorem 4.4] Let $\mathcal{F}$ be a finite multi-set of Boolean constraint functions of arity k, and $S_{\mathcal{F}}(y)$ be its sensitivity polynomial. Let $\gamma_{\mathcal{F}}$ and $\Phi_{\mathcal{F}}$ be functions defined on (1,2] by*

- *$\gamma_{\mathcal{F}}(y) = \ln\left(\frac{y}{2(y-1)} \right) \cdot \frac{S_{\mathcal{F}}(y)}{S'_{\mathcal{F}}(y)}$*

- *$\Psi_{\mathcal{F}}(y) = (y-1)\ln(y-1) - y\ln(\frac{y}{2}) + \gamma_{\mathcal{F}}(y) \cdot \ln\left(\frac{S_{\mathcal{F}}(y)}{2^k \cdot |\mathcal{F}|} \right)$*

The probability of satisfiability of a random $\mathcal{F}$-formula with n variables and $c \cdot n$ constraints tends to 0 for every $c > c^*_{\mathcal{F}}$, where $c^*_{\mathcal{F}} = \gamma_{\mathcal{F}}(y^*_{\mathcal{F}})$ and $y^*_{\mathcal{F}}$ is the unique number such that for every $y < y^*_{\mathcal{F}}$, $\Psi_{\mathcal{F}}(y) < 0$.

It is proven in [4] that this threshold is always strictly smaller than the usual threshold with the FMM.

The theorem can be applied to any CSP. The equation stated for $\Psi_{\mathcal{F}}$ has to be solved, with k the number of variables in each clause. We use the theorem together with the calculation of the sensitivity polynomial for shapes in order to obtain upper bounds for the critical values of different SAT problems for shapes. The solutions are computed with a small Maple program. Several experiments with different shapes are included in Chapter 4.

Chapter 4

Results and Experiments

In this chapter, we show the sensitivity polynomials for several shapes; some taken from [18] and some new ones. We compute and compare two upper bounds for each shape. The first upper bound is obtained by the first moment method (FMM), the second one by the method of locally maximal solutions (LMSM). We consider absolute and relative improvements of the new method over the first moment method and discuss the limitations.

4.1   Computing New Upper Bounds

In order to avoid the manual computation of the sensitivity polynomial, we use a program (coded in `C`) to perform the calculation. This calculation is based on Lemma 3.3.2. The equations in Figure 4.1 represent the polynomials for several shapes. Since our focus is disjunctive shapes, the computation of their sensitivity polynomial is sufficient here.

On "usual" computers, the size of an integer is restricted to 32 or 64 bits. In order to be able to handle shapes with more than 64 variables (having more than 2^{64} different interpretations), the bignum package `gmp`[1] has been used. With such a package, it is possible to deal with integers of arbitrary length.

In order to compute the upper bounds by the first moment method (FMM) and the locally maximal solutions method (LMSM), we use a `MAPLE` program provided by H. Daudé. The parameter that represents the upper bound obtained by the FMM is called dd, while the parameter obtained with the LMSM is represented by cc. In Table 4.1, we compare dd and cc for the shapes from Figure 4.1.

---

[1]Information can be found on the project web page http://gmplib.org.

$$\begin{aligned}
S_{\langle 3 \rangle}(y) &= 4 + 3\,y \\
S_{\langle 4 \rangle}(y) &= 11 + 4\,y \\
S_{\langle 5 \rangle}(y) &= 26 + 5\,y \\
S_{\langle 3,2 \rangle}(y) &= 10 + 27\,y^2 \\
S_{\langle 2,4,2 \rangle}(y) &= 6911 + 2800\,y + 8400\,y^2 + 11200\,y^3 + 5600\,y^4 \\
S_{\langle 6,3 \rangle}(y) &= 43653 + 100842\,y^3 \\
S_{\langle 2,2,2,2 \rangle}(y) &= 2815 + 4968\,y^2 + 14904\,y^4 \\
S_{\langle 2,2,3,2 \rangle}(y) &= 2419561 + 2945160\,y^2 + 3975966\,y^4 \\
S_{\langle 3,3,2 \rangle}(y) &= 104709 + 24642\,y + 49284\,y^2 + 32856\,y^3 \\
S_{\langle 2,2,4,2 \rangle}(y) &= 1251321583 + 1010463984\,y^2 + 814403808\,y^4 \\
S_{\langle 2,5,3 \rangle}(y) &= 315163377 + 122580480\,y + 183870720\,y^2 + 137903040\,y^3 + \\
&\quad 51713640\,y^4 + 7757046\,y^5 \\
S_{\langle 2,2,2,2,2 \rangle}(y) &= 2245537151 + 482779360\,y + 625072224\,y^2 + 284585728\,y^3 + \\
&\quad 142292864\,y^4 \\
S_{\langle 2,2,2,2,3 \rangle}(y) &= 254296788108031 + 14873496696000\,y + 7436103021000\,y^2 + \\
&\quad 1393906320000\,y^3 + 261357435000\,y^4 \\
S_{\langle 3,3,3,3 \rangle}(y) &= 9690944106639854579830 + 10720444695878647674252\,y^3 + \\
&\quad 7163206228609823 6732502\,y^6 + \\
&\quad 159544138728127890904209\,y^9
\end{aligned}$$

Figure 4.1: Sensitivity polynomials $S_{\mathcal{S}}(y)$ for disjunctive shapes $\mathcal{S}$.

$\mathcal{S}$	v	$w^1(\mathcal{S})$	dd	cc
$\langle 3 \rangle$	3	7	5.1908930697	4.6424761577
$\langle 4 \rangle$	4	15	10.7400536663	10.2168796332
$\langle 5 \rangle$	5	31	21.8323023417	21.3202500034
$\langle 3, 2 \rangle$	6	37	1.2649475142	1.0436294188
$\langle 2, 4, 2 \rangle$	16	34911	1.1005871282	1.0071276700
$\langle 6, 3 \rangle$	18	144495	1.1636838197	1.0713962003
$\langle 2, 2, 2, 2 \rangle$	16	22687	0.6534146441	0.5620828127
$\langle 2, 2, 3, 2 \rangle$	24	9340687	1.1835681431	1.1201671090
$\langle 3, 3, 2 \rangle$	18	211491	3.2282718641	3.1418610777
$\langle 2, 2, 4, 2 \rangle$	32	3076189375	2.0768310115	2.0311970470
$\langle 2, 5, 3 \rangle$	30	818988303	2.5592963773	2.5044459763
$\langle 2, 2, 2, 2, 2 \rangle$	32	3780267327	5.4300936720	5.3866728906
$\langle 2, 2, 2, 2, 3 \rangle$	48	278261651580031	60.3698019631	60.3476830278
$\langle 3, 3, 3, 3 \rangle$	81	251587589816744629890793	0.3063168922	0.2773786031

Table 4.1: Upper bounds obtained by the FMM (dd) and the LMSM (cc) for the shapes $\mathcal{S}$ with v variables and $w^1(\mathcal{S})$ satisfying assignments. All numbers are rounded to 10 decimals.

4.2   Comparing the LMSM with the FMM

A further interesting aspect is a more detailed comparison between both methods, as well as the analysis of the behaviour of the phase transition in general with respect to the operators $\wedge$ and $\vee$. Experiments show the evolution of phase transition when the size of the formulas is increased by modifying one parameter.

Figure 4.2 shows both upper bound for a shape $\langle k \rangle$. The chart shows the exponential growth of the upper bound when k is increasing.

In Figure 4.3, the gain of the LMSM with respect to the FMM shows that the larger k is, the smaller the difference between the FMM upper bound and the LMSM upper bound becomes. Observe that the difference decreases until it reaches an (experimental) asymptotic value in the axis Y, namely $Y = 0.5$. Figure 4.4 shows the relative gain with respect to the FMM. For smaller k, there is a considerable improvement with the LMSM. With increasing k, the relative improvement is decreasing.

Let us turn our attention to shapes with more structure. Figures 4.5 and 4.6 show the upper bound, the absolute improvement and the relative improvement of the LMSM over the FMM for the shapes of the form $\langle k_1, 2 \rangle$. In these shapes, two levels of operation nesting are present. In this case, the arity of the top-most $\vee$ is increased.

With the shapes of the form $\langle 2, k_2 \rangle$, we investigate the behaviour of the shapes when the number of subformulas of the conjunction (and not those of the disjunction) is increased. In this case, the upper bound decreases for larger

k_2 (Figure 4.7) and we also see that the relative improvement increases (Figure 4.8). This is in contrast to the two former cases.

Results for the shapes of the form $\langle 2, 2, k_3 \rangle$ are represented in Figures 4.9, 4.10 and 4.11. There are three levels of nesting in these shapes, and the number of subformulas of the disjunction is increased in this case. However, the disjunctive operator whose arity is changing is not the top-most operator. The experiments in Figure 4.10 show that the absolute improvement goes asymptotically down to a constant $Y = 0.125$. In our experiments, besides these and $\langle k \rangle$-shapes, no other shapes show a tendency to decrease the absolute difference among the FMM upper bound and the LMSM upper bound to an asymptotic value $Y = c$, $c > 0$.

The next results presented are for shapes of the form $\langle 2, k_2, 2 \rangle$. The connective that increases its arity is the conjunction. In this case, however, the arity of the middle connective is changing. The upper bound decreases (Figure 4.12) and the relative improvement increases (Figure 4.13), similarly to the shapes $\langle 2, k_2 \rangle$. From the experimental data, one possible conclusion is that the number of locally maximal solutions decreases considerably if the arity of the conjunction connectives is increased. Therefore, the upper bound based on the LMSM improves when the number of conjuncts is increased.

In the shapes of the form $\langle 2, 2, k_3, 2 \rangle$, we are changing the number of disjunctives in a level that is neither the upper nor the lowest in the nesting. The results for such shapes are shown in Figures 4.14, 4.15 and 4.16. The results are very similar to those for the shapes $\langle 2, 2, k_3 \rangle$ and $\langle k \rangle$. However, the difference between dd and cc seems to approach zero.

Finally, we present shapes of the form $\langle 2_1, ..., 2_d \rangle$ for varying d. In this case, the increment is on the depth of nesting. The results show two different tendencies, one for shapes with d being an even number and one for d being an odd number. Shapes with its last operator being an $\vee$ behave quite different, its upper bound grows with higher depths. Shapes with $\wedge$ as its last operator have a decreasing upper bound when the depth increases.

The experiments shown here give some insight and ideas for further research in order to generalize results for arbitrary shapes.
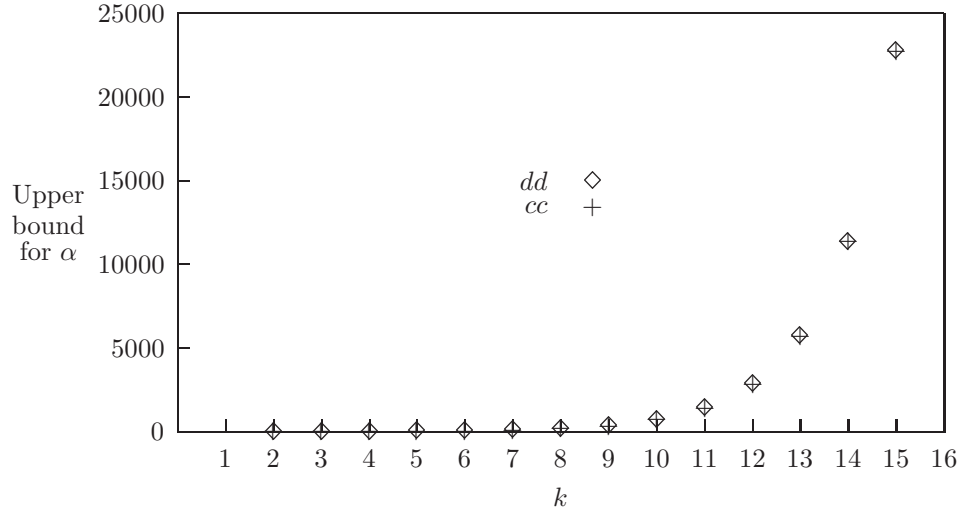
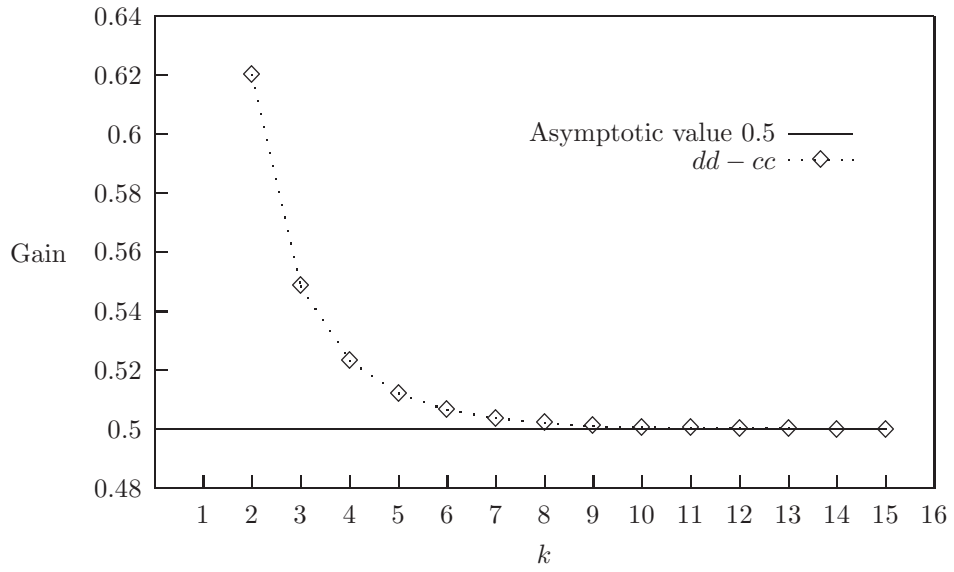Figure 4.2: Upper bound with the FMM (dd) and the LMSM (cc) for the shapes $\langle k \rangle$.

Figure 4.3: Absolute difference between upper bounds with the FMM (dd) and the LMSM (cc) for the shapes $\langle k \rangle$.
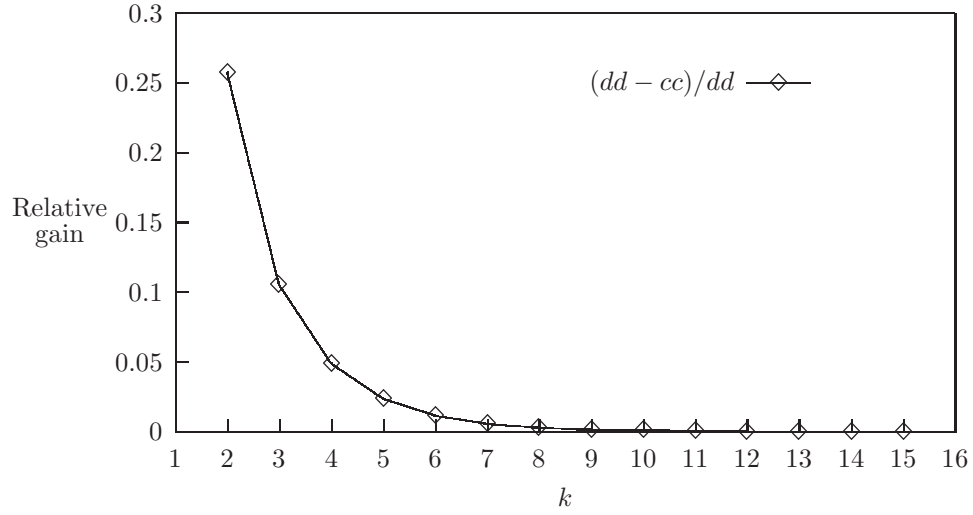
39

Figure 4.4: Relative gain of the LMSM (cc) with respect to the FMM (dd) for the shapes $\langle k \rangle$.
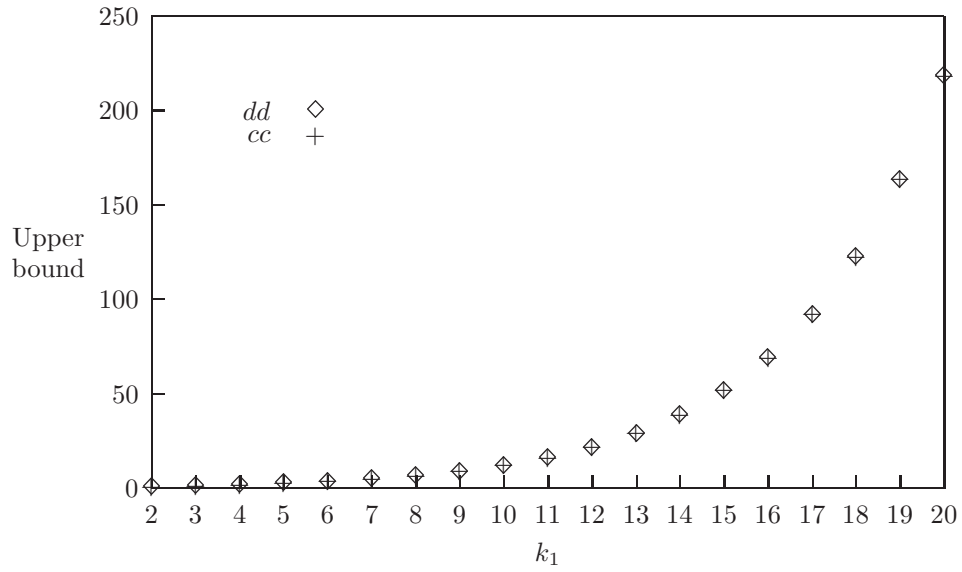

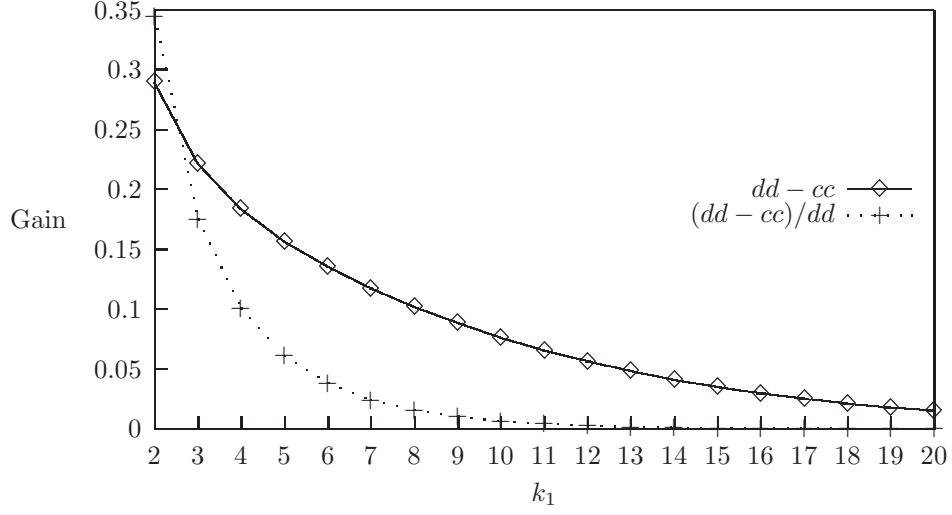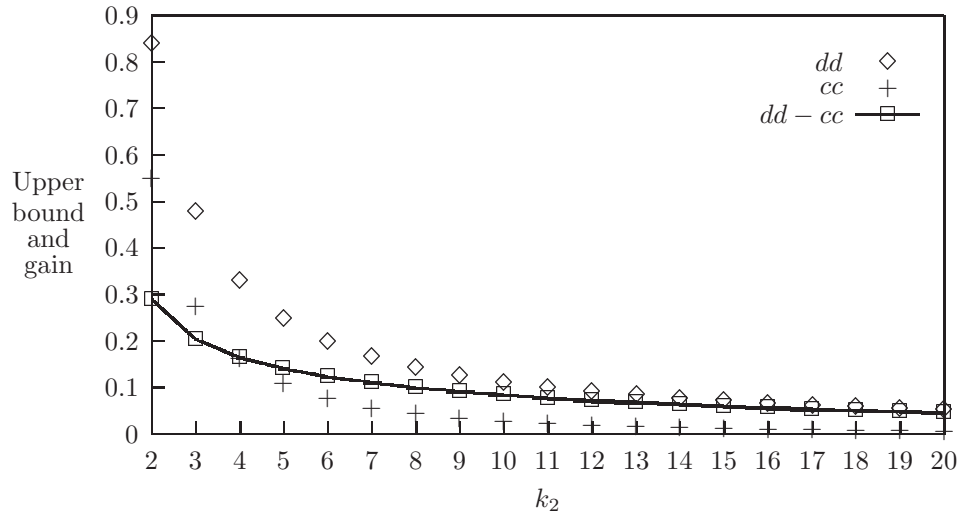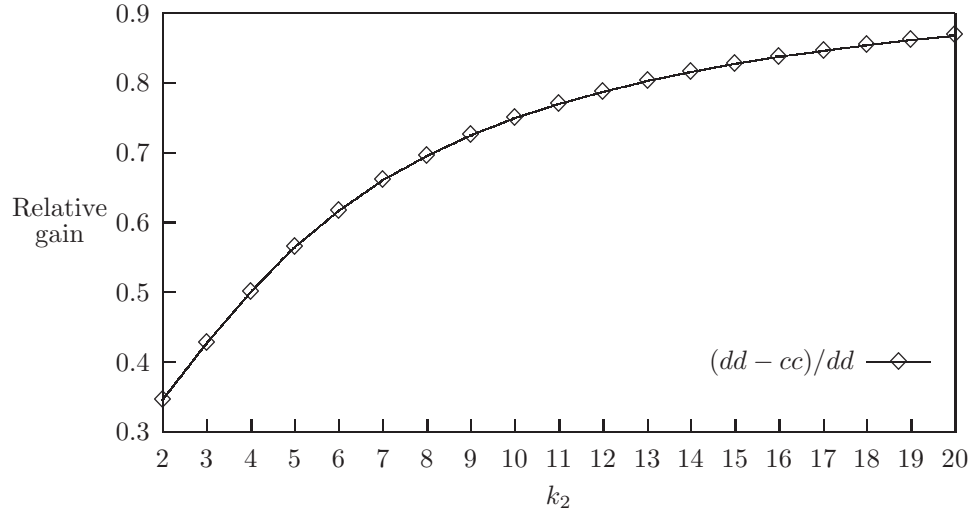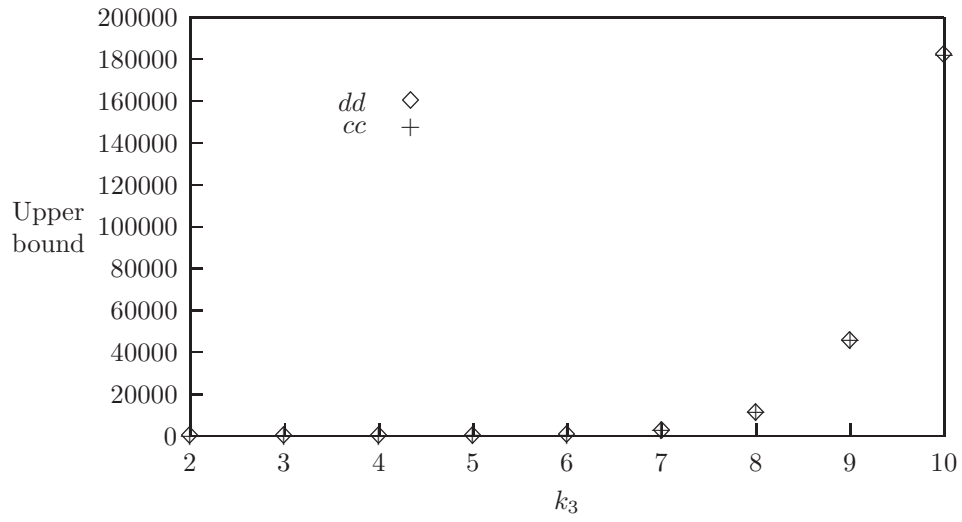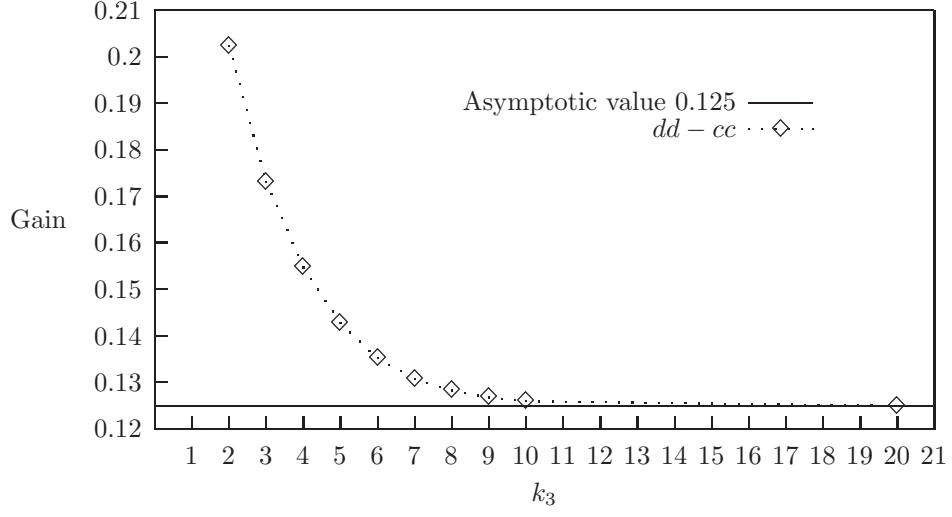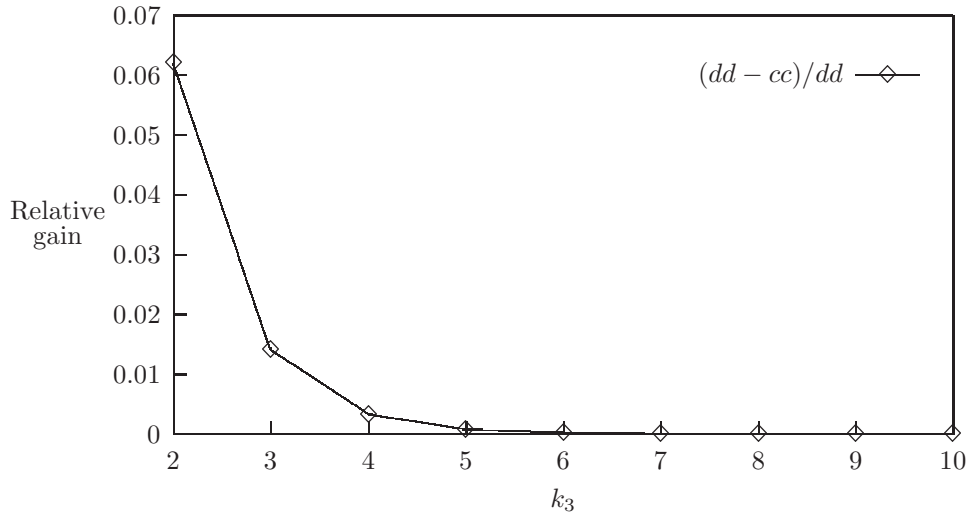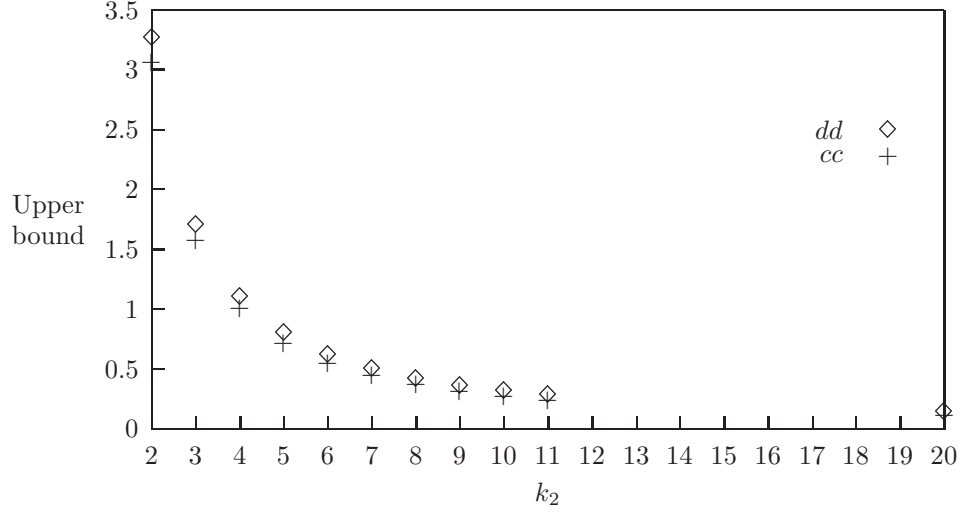Figure 4.5: Upper bound with the FMM (dd) and the LMSM (cc) for the shapes $\langle k_1, 2 \rangle$.

Figure 4.6: Absolute and relative difference between upper bounds with the FMM (dd) and the LMSM (cc) for the shapes $\langle k_1, 2 \rangle$.

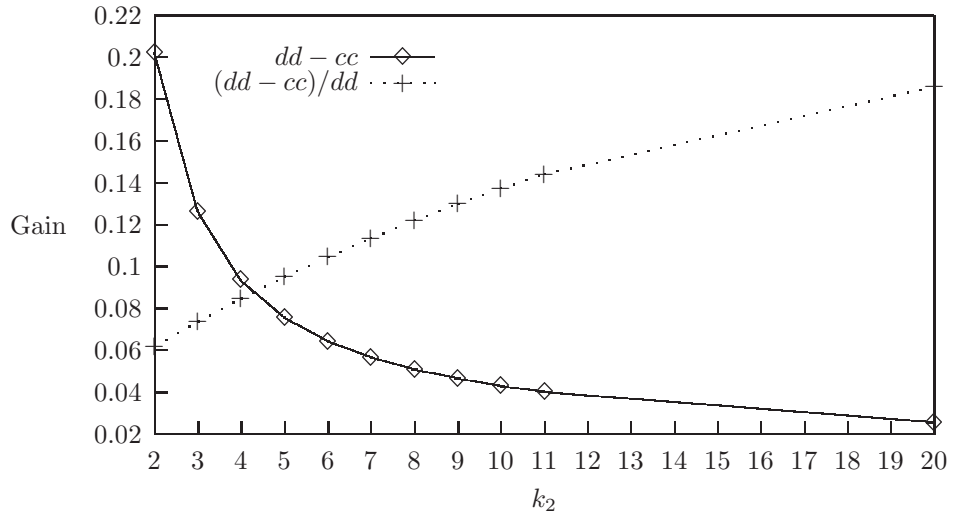Figure 4.7: Upper bounds with the FMM (dd) and the LMSM (cc) for the shapes $\langle 2, k_2 \rangle$, and the absolute differences between them.

41

Figure 4.8: Relative gain of the LMSM (cc) with respect to the FMM (dd) for the shapes $\langle 2, k_2 \rangle$.
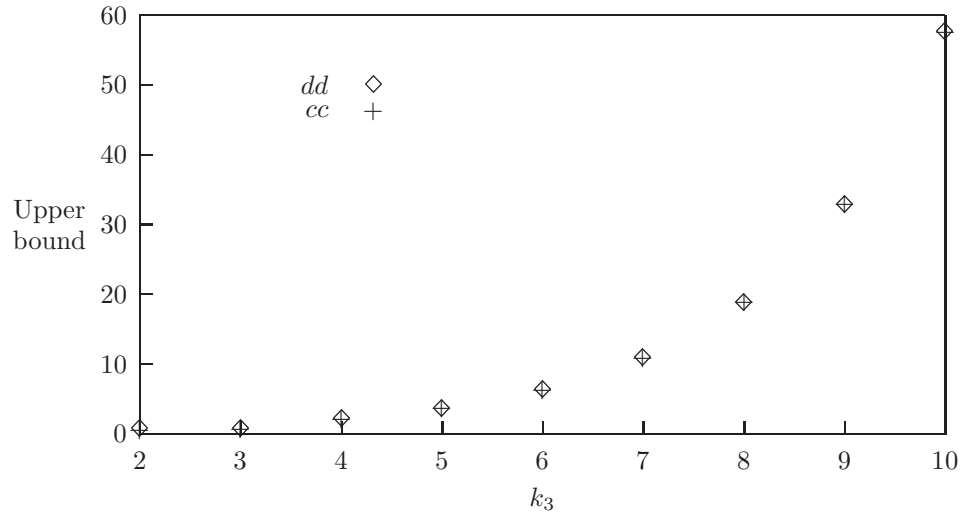

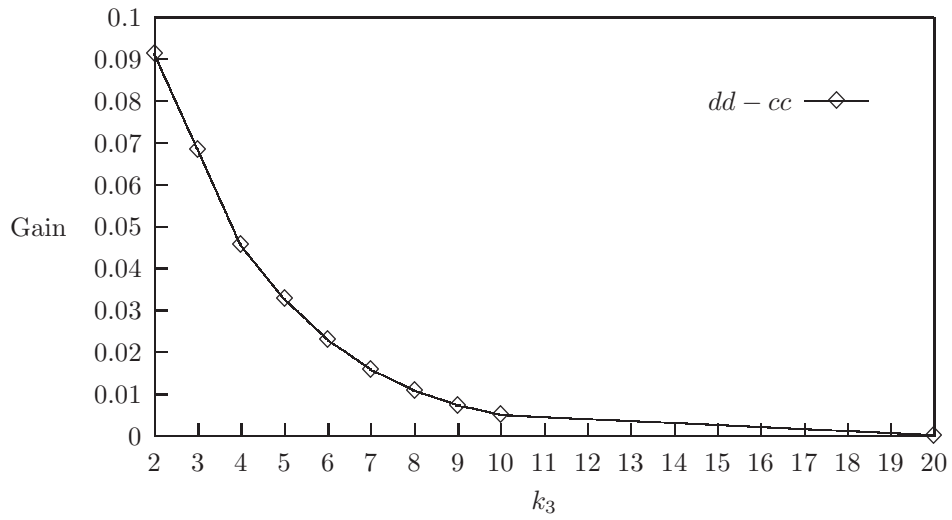Figure 4.9: Upper bound with the FMM (dd) and the LMSM (cc) for the shapes $\langle 2, 2, k_3 \rangle$.

Figure 4.10: Absolute difference between upper bounds with the FMM (dd) and the LMSM (cc) for the shapes $\langle 2, 2, k_3 \rangle$.

Figure 4.11: Relative gain of the LMSM (cc) with respect to the FMM (dd) for the shapes $\langle 2, 2, k_3 \rangle$.

Figure 4.12: Upper bound with the FMM (dd) and the LMSM (cc) for the shapes $\langle 2, k_2, 2 \rangle$.

Figure 4.13: Gain and relative gain between of the LMSM (cc) with respect to the FMM (dd) for the shapes $\langle 2, k_2, 2 \rangle$.

Figure 4.14: Upper bound with the FMM (*dd*) and the LMSM (*cc*) for the shapes $\langle 2, 2, k_3, 2 \rangle$.

Figure 4.15: Absolute difference between upper bounds with the FMM (*dd*) and the LMSM (*cc*) for the shapes $\langle 2, 2, k_3, 2 \rangle$.
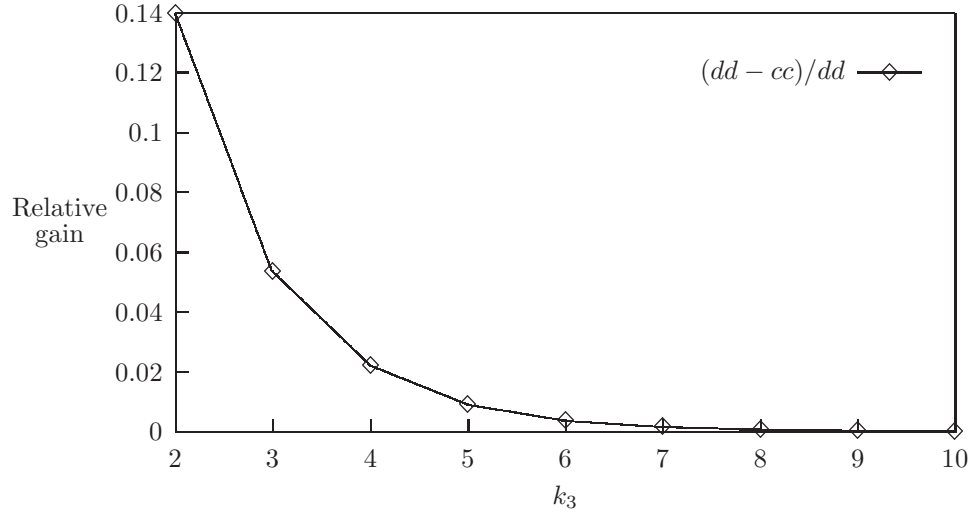
Figure 4.16: Relative gain of the LMSM (cc) with respect to the FMM (dd) for the shapes $\langle 2, 2, k_3, 2 \rangle$.
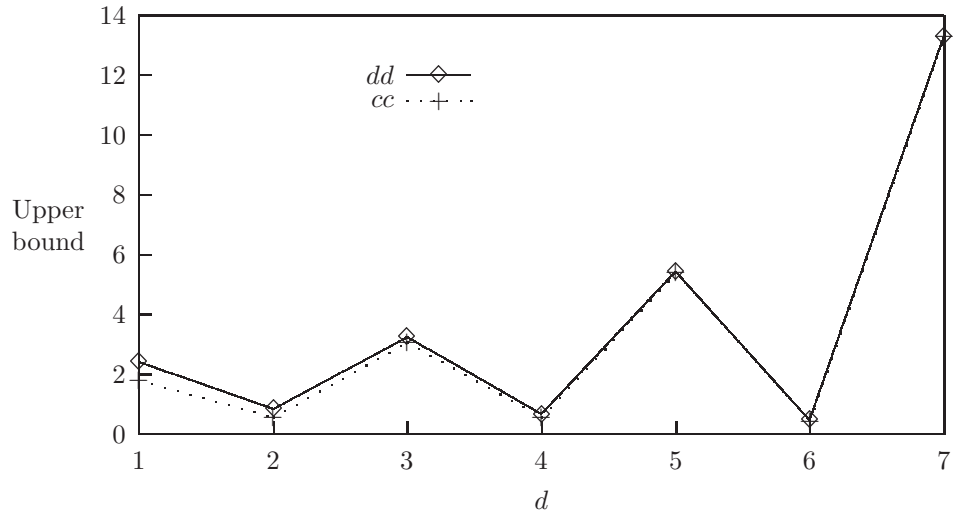

Figure 4.17: Upper bound with the FMM (dd) and the LMSM (cc) for the shapes of the form $\langle 2_1, ..., 2_d \rangle$ with depth d.
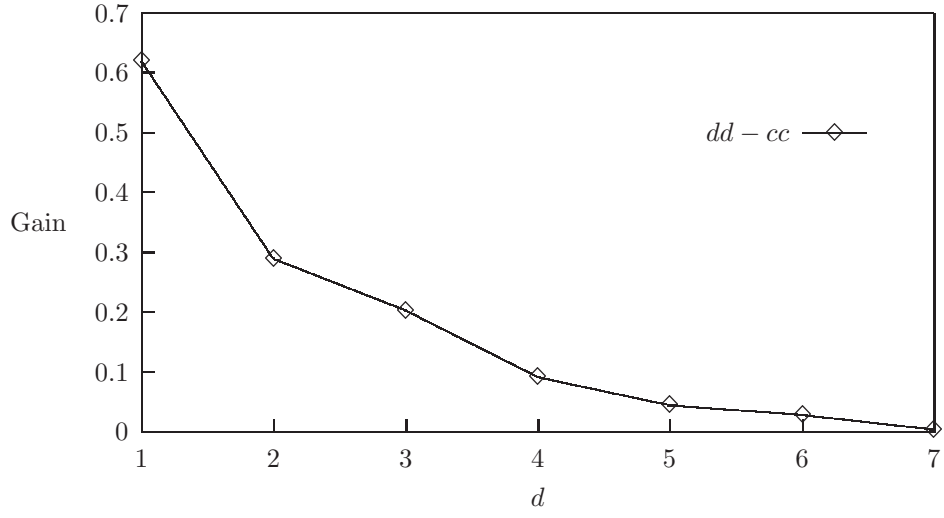
Figure 4.18: Absolute difference between upper bounds with the FMM (dd) and the LMSM (cc) for the shapes of the form $\langle 2_1, .., 2_d \rangle$ with depth d.
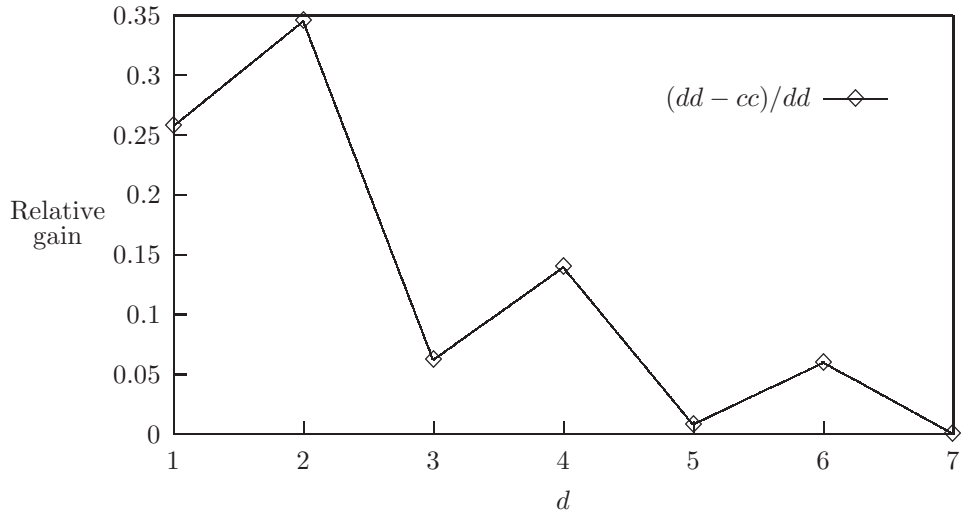

Figure 4.19: Relative gain of the LMSM (cc) with respect to the FMM (dd) for the shapes $\langle 2_1, ..., 2_d \rangle$ with depth d.

Chapter 5

Conclusions

In this thesis, we considered a generalization of the well known K-SAT problems. In such K-SAT problems, the formulas consist of a conjunction (or set) of clauses. In our more general framework, the formulas consist of conjunctions (or sets) of balanced shapes. These shapes have the following properties:

- The only connectives occurring in the shapes are $\wedge$ and $\vee$.

- If the shape is represented as a tree, the tree is balanced and the connectives are alternating along each branch in the tree.

- The starting connective, i.e., the top-most in the tree, is $\vee$.

- The leafs represent variables and each variable occurs only once in the shape.

As in the K-SAT problem, we consider a set of randomly chosen shape instances. Since we are interested in shapes of the form $\langle k_1, ..., k_d \rangle$, we call the corresponding random satisfiability problem, random $\langle k_1, ..., k_d \rangle$-SAT. As in random K-SAT, the random $\langle k_1, ..., k_d \rangle$-SAT problem shows a SAT/UNSAT threshold observed with the help of the ratio α between the number of clauses and the number of variables. First upper bounds computed with the first moment method were taken from the literature [18].

With a refined method, developed in [4], we improved all the upper bounds for different SAT problems for different shapes from [18]. This method uses as an input a sensitivity polynomial. Given a shape with v variables, this polynomial represents the distribution of the solutions in the $\{0, 1\}^v$ hypercube. The core of the method is to reduce the number of solutions to those that are "locally maximal". The last step of the method is to solve an equation using the sensitivity polynomial, a task that can be solved with the help of software for mathematics.

Experimental results with shapes of different forms have produced new ideas that can guide to further research. A comparison of the two upper bounds,

obtained by the traditional first moment method and the new method, was presented.

The behaviour of shapes has shown to be different when we modify parameters of the shape like the number of disjunctives, conjunctives, or the depth. In summary, we have the following observations.

- If the arity of a conjunction increases, then the upper bound tends to decrease.

- If the arity of a disjunction increases, then the upper bound tends to increase.

- If the depth is increased, then the upper bound increases for odd depth and slightly decreases for even depth.

There are several lines of research for future work. One of them is to find properties of a solution which allow to consider a smaller set of solutions (like the locally maximal solutions with respect to the total number of solutions). The decrease of the number of considered solutions would result in smaller upper bounds. Another issue to be solved is a characterization that allows to find lower bounds for the critical value for shapes. This characterization has to be general enough to be applied to any shape. Finally, a further task is to perform more experiments in order to obtain empirical critical values for different shapes.

Bibliography

[1] D. Achlioptas. Lower bounds for random 3-sat via differential equations. *Theoretical Computer Science*, 265(1-2):159–185, 2001.

[2] V. Chvátal and B. Reed. Mick gets some (the odds are on his side). In *FOCS*, pages 620–627. IEEE, 1992.

[3] S. Cook. The complexity of theorem-proving procedures. *Third Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971. STOC 1971, ACM Press.

[4] N. Creignou, H. Daudé, and O. Dubois. Expected number of locally maximal solutions for random boolean CSPs. In *Conference on Analysis of Algorithms 2007*, Juan-les-Pins, France, June 2007. Discrete Mathematics and Theoretical Computer Science.

[5] O. Dubois and Y. Boufkhad. A general upper bound for the satisfiability threshold of random r-SAT formulae. *J. Algorithms*, 24(2):395–420, 1997.

[6] O. Dubois, R. Monasson, B. Selman, and R. Zecchina. Editorial. *Theoretical Computer Science*, 265(1–2), 2001.

[7] P. Erdős and A. Rényi. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kut. Int. Kőzl.*, 5:17, 1960.

[8] J. Franco. Results related to threshold phenomena research in satisfiability: lower bounds. *Theoretical Computer Science*, 265:147–157, 2001.

[9] J. Franco and M. Paull. Probabilistic analysis of the Davis-Putnam procedure for solving the satisfiability problem. *Discrete Applied Mathematics*, 5(1):77–87, 1983.

[10] E. Friedgut. Necessary and sufficient conditions for sharp thresholds of graph properties, and the K-SAT problem. *J. Amer. Math. Soc.*, 12:1017–1054, 1999.

[11] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[12] A. Goerdt. A threshold for unsatisfiability. *17th International Symposium of Mathematical Foundations of Computer Science*, 629:264–275, 1992.

[13] A. K. Hartmann and M. Weigt. *Phase Transition in Combinatorial Optimization Problems*. WILEY-VCH, 2005.

[14] S. Kirkpatrick and B. Selman. Critical behavior in the satisfiability of random Boolean expressions. *Science*, 264(5163):1297–1301, 27 May 1994.

[15] L. M. Kirousis, E. Kranakis, D. K., and Y. C. Stamatiou. Approximating the unsatisfiability threshold of random formulas. *Random Structures and Algorithms*, 12(3):253–269, 1998.

[16] D. G. Mitchell, B. Selman, and H. J. Levesque. Hard and easy distributions for SAT problems. In Paul Rosenbloom and Peter Szolovits, editors, *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 459–465, Menlo Park, California, 1992. AAAI Press.

[17] M. Molloy. The probabilistic method. In M. Habib, C. Diarmid, J. Ramirez-Alfonsin, and B. Reed Editors, editors, *Probabilistic Methods for Algorithmic Discrete Mathematics*, pages 1–35. Springer, 1998.

[18] J. A. Navarro Pérez and A. Voronkov. Generation of hard non-clausal random satisfiability problems. In *Proceedings of the Twentieth National Conference on Artificial Intelligence and the Seventeenth Annual Conference on Innovative Applications of Artificial Intelligence*, Pittsburgh, PA, July 2005. AAAI Press.

[19] P. Slavík. *Approximation Algorithms for set cover and related problems*. PhD thesis, New York University, Buffalo, April 1998.