



**KTH Computer Science  
and Communication**

# **On State Space Structure and Average Case Complexity in Random K-SAT Problems**

JOHN ARDELIUS

Licentiate Thesis  
Stockholm, Sweden 2008

TRITA-CSC-A 2008:08

ISSN-1653-5723

ISRN-KTH/CSC/A-08/08-SE

ISBN-978-91-7178-971-6

KTH School of Computer Science and Communication

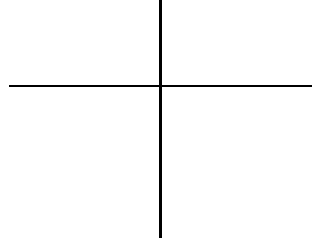
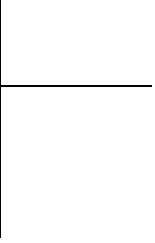
SE-100 44 Stockholm

SWEDEN

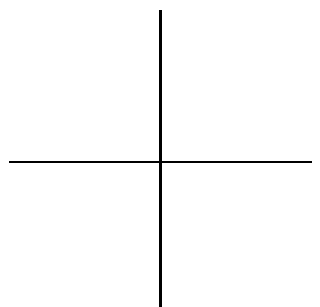
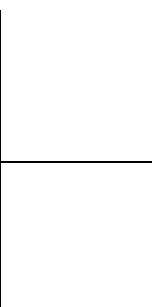
Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie licentiatexamen i datalogi onsdagen den 14 maj 2008 klockan 13.00 i Sal FR112 Kungl Tekniska högskolan, Roslagstullsbacken 23, Stockholm .

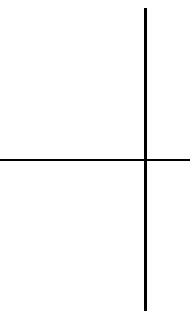
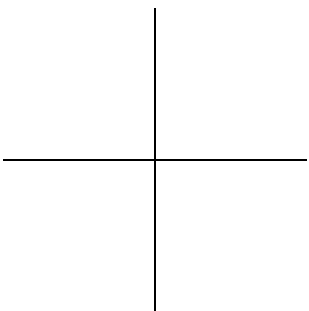
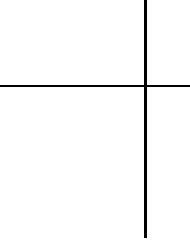
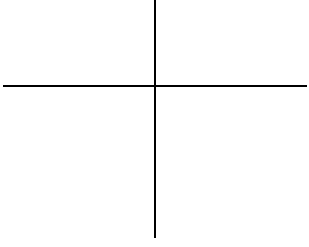
© John Ardelius, maj 2008

Tryck: Universitetsservice US AB



*To my grandparents*





## Abstract

This thesis gives an introduction to a currently active area in the cross-section between theoretical computer science and theoretical physics. In the last ten years it has been suggested that critical behaviour, usually seen in models from condensed matter physics, may be responsible for the intractability of NP complete computation problems. This would suggest a very deep connection between the two fields on the most fundamental level. How deep this connection really is is subject to ongoing research as well as the topic of this thesis. Some of the conjectures from the physics community regarding computational hardness in certain problem classes has turned out to be wrong or misinterpreted but the gained interest in both fields has promising potential in moving the research frontier forward.

The material presented in this thesis is the result of nearly two years work in trying to clarify how the results from physics can be interpreted in the language of actual computation problems.

## Sammanfattning

Denna avhandling ger en introduktion till ett mycket aktivt forskningsområde i gränslandet mellan teoretisk datalogi och teoretisk fysik. Under de senaste tio åren har det framkommit forskningsresultat som pekar på att kritiska fenomen, vanligen hemmahörande i modeller från teoretisk materialfysik, kan vara nyckeln till att förstå varför NP fullständiga problem är så svåra att lösa. Detta skulle innebära en mycket djup och fundamental koppling mellan de bägge områdena. Hur djup denna koppling verkligen är är temat i mycket av pågående forskning såväl som ämnet för denna avhandling. Vissa förutsägelser från den teoretiska fysiken har visat sig felaktiga eller feltolkade men det ökade intresset för dylika frågor inom bägge forskningsområden ger hopp om att tillsammans kunna flytta fram forskningsfronten.

I would like to thank my supervisors Erik Aurell, Supriya Krishnamurthy and Seif Haridi for giving me the opportunity to work in a very fascinating and active field of research. Thanks also to SICS and Sverker Jansson for providing a way to work full time with these topics over the past two years. Special thanks to Frej Drejhammar for all invaluable help with debugging c-code, Linux tips and trix and more. Thanks to the EVERGROW project for providing both the computational resources which made most of the results in this thesis possible and the opportunity to meet many interesting people in this and related research areas. At last I would like to thank my family Regina and Hugo for putting up with me in stressful times.

# Contents

<b>Contents</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 General background . . . . .	2
1.2 SAT solving techniques . . . . .	6
1.3 A physics interpretation . . . . .	13
1.4 Solution space structure . . . . .	16
<b>2 Papers and contributions</b>	<b>21</b>
2.1 Paper 1 . . . . .	21
2.2 Paper 2 . . . . .	22
2.3 Paper 3 . . . . .	23
2.4 Paper 4 . . . . .	24
2.5 Paper 5 . . . . .	25
2.6 Conclusion . . . . .	26
<b>Bibliography</b>	<b>27</b>
<b>I Papers</b>	<b>33</b>



# Chapter 1

## Introduction

During the past ten year it has been suggested that there is a relation between computational hardness and solution space structure in combinatorial optimization problems. By using methods traditionally used to treat physical models of disordered spin-glass materials, structural properties of the space of solutions in certain problems is found to have a very non-trivial structure. In spin-glasses the complex structure of energetic ground states results in a very slow relaxation process also seen in other amorphous materials. Interpreting an algorithm trying to solve an optimization problem as such a process has led people to believe that algorithmic hardness may be due to critical phenomenons in the space of solutions. This new connection between physics and computer science has inspired people to look at computational complexity in a new way [41].

In this Licentiate thesis I will describe this connection from both a computer science and a physicists perspective with a brief overview of related topics. I will start by giving an introduction to the subject of complexity theory in general, with emphasis on average case complexity.

Then follows an overview of the current statistical physics view of the solution space structure in random satisfiability problems together with a section describing what rigorous results that has been obtained.

I will try to provide an as complete background as possible to understand the context in which the work presented here is done. However, since the field is a crossection between various parts of computer science reaching through message passing methods, satisfiability testing, complexity theory as well as a major part of cutting edge theoretical physics I will only introduce each of the fields on a brief level in order to make the context of this work clear. The focus has been to try to communicate an intuitive picture of the problems and the methods at hand with as few equations and formal notation as possible. For a more complete review of any subject please follow the notes in the bibliography.

## 1.1 General background

This section provides a general introduction to related topics in computer science and physics. The intention is to provide definitions and descriptions of concepts considered common knowledge in the respective fields which will be used throughout the rest of the thesis. Claims that are stated without proofs are considered well known and details can be found in the references. These are complemented by references to important key papers which can be consulted for a more complete survey. A detailed description of the subject of complexity theory can be found in [43, 4, 24].

### Complexity theory

The study of how intrinsically complex computational tasks are in computer science is called *complexity theory*.

In complexity theory one wants to classify problems according to the amount of resources required to solve them. The most common computational resources are the time it takes to solve the problem and the amount of working memory needed in the process. The term *problem* in this context is however more precise than what we normally mean by a the term in everyday language. In order to make a proper definition I will start by giving a few well known examples. The first one is the traveling salesman problem (TSP). This is the problem of finding a path between a set of cities on a map such that the time it takes to visit them all is minimized. The input to the problem is a given start position and the travel time between all pairs of cities. Another example is the *graph coloring problem* in which one has the task to, using  $q$  colors, color the vertex set of a given graph in different colors such that no edges connects two vertexes with the same color. In practical applications one is often interested in different kinds of *packing problems*, also known as *knapsack* problems which are concerned with the task to select a subset of objects from a certain set such that constraints (price, weight etc) are fulfilled while space is minimized. All these three examples are problems in the computational complexity terminology where as an *instance* of a problem specifies explicit input variables such as the map in TSP or the graph and number of colors in the coloring problem. A problem is then how to solve the set of all possible instances.

One distinction made between types of problems in complexity theory is between *optimization* and *decision* problems. In optimization problems one wants to get an optimal solution among all valid ones to the problem. In decision problems one wants to find out whether or not (yes or no) a valid solution exist to the given input. Examples of optimization problems are: 'What is the shortest path between two nodes in a network?' while decision problems provides answers to questions like 'Is this equation solvable?'. Optimization problems can be reduced to decision problems in polynomial time. In the TSP for instance, this is done by starting from a worst case path length  $X$  (the sum of all edges in the graph) and then ask

questions of the form 'Is there a path between city A and city B which visits all other cities on the map and is shorter than  $X$ ?'. If the answer is yes than just ask the same question again with the value  $X/2$ . If the answer is no than the value  $X$  is already optimal. Iterating this procedure will eventually give the optimal value  $X^*$  using a number of questions polynomial in system size. The method formally known as *binary search* can be used to reduce any optimization problem to a decision problem.

A way of looking at a computational problem is how computational resources needed to solve the problem depends on the size of input data. The size of input data is the number of bits sent as input to the problem. Using this relation we say that a problem is in the  $P$  or polynomial-time class if the time it takes to solve the problem is upper bounded by a polynomial function in the size of input data. This applies to all instances of the problem. In extending the set of problems in  $P$  to include all problems for which a solution can be verified (of falsified) in polynomial time we get another very central and important complexity class namely the Non-deterministic Polynomial problems (NP). For problems in this set the time it takes to solve an instance can not guaranteed to be upper bounded by a polynomial function in size of input data but once we have a solutions candidate it can be verified in polynomial time. All of the three examples given above belongs to  $NP$  because a solution to any one of them can be verified in polynomial time.

A key concept in the systematic classifications of computational problems in complexity classes is the concept of *reductions*. We say that a problem A can be reduced to B if solving B implies that we can automatically solve A. Therefore A cannot be harder (in some sense) than B. This notion provides a way for us to order the problems by complexity.

By combining the concepts of reduction and NP we can define another very important complexity class namely the NP-complete problems (NPC). These are problems which belong to NP and has the property that every other problem in NP is reducible to it in polynomial time. This means that if we have an efficient way to solve one NPC problem in polynomial time we can solve all of them in polynomial time. The major open question in complexity theory is whether or not the problems that can be verified in polynomial time, NP, are also solvable in polynomial time P. That is if  $P=NP$ .

Methods for solving computational problems can be divided into *exact* algorithms and *heuristic* algorithms. Strictly speaking an exact algorithm is a logical sequence of steps that one follows in order to solve a problem where as a heuristic algorithm is a more general solving approach which might not lead to a solution.

In practice, if a problem is shown to be NP-complete, it means that since the running time will grow super polynomial in in data size one is restricted to treating small systems or use approximate methods.

$$F = (x \vee \neg y \vee z) \wedge (x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee z) \wedge (\neg x \vee y \vee \neg z)$$

Figure 1.1: Example of a 3-SAT formula with  $N = 3$  variables  $[x, y, z]$  and  $M = 4$  clauses. It is in this case easy to find a satisfying assignment for instance  $x = 1$ ,  $y = 0$  and  $z = 0$ .

### The satisfiability problem

The very first problem shown to be NP-complete was the Boolean satisfiability problem (SAT) by the Cook-Levin theorem in early 70's [12, 32]. The SAT problem is a decision problem where each instance contains  $N$  Boolean variables  $[0|1]$  related through a number of NOT, OR and AND logical operators together with parentheses. The problem is considered solved and the formula is satisfied if one finds a configuration of the  $N$  variables such that all logical constraints, posed by the operators, are fulfilled.

A convenient way of looking at a SAT instance is to pose the problem in so called conjunctive normal form (CNF). In this format the variables occurs either negated or un-negated (logical NOT operator denoted  $\neg$ ). A variable together with its sign is denoted a *literal*. One then groups the literals into clauses by separating them by logical OR operators. A clause is then satisfied if one or more of the contained literals are. The clauses are finally separated by logical AND operators which creates a conjunction of disjunctions hence the name of the format. Every logical formula can be cast into a corresponding CNF formula and the problem of satisfying such an instance is called CNF-SAT which is also a NP-complete problem.

By restricting the format even further one can create  $k$ -satisfiability (K-SAT) problems. In these problems the number of literals in each clause is fixed to  $K$ . K-SAT is, unlike general SAT, only NP-complete if  $K$  is greater or equal to three. For  $K=2$  on the other hand the problem can, even in the worst case, be solved in polynomial time. Due to historical reasons the 3-SAT problem has been a very central problem in complexity theory. It is one of Karp's famous 21 problems [26] and one often tries to reduce new problems to 3-SAT in order to show that they are NP-hard. It has also been regarded, due to its inherent hardness, as the standard benchmark problem in evaluating new algorithms and approximate solutions methods.

Many real world industrial problems such as planning [28], scheduling [8] and hardware verification [53] are or can be transformed to various forms of SAT problems which further motivates their central role.

In 1972 Richard M. Karp extended the work of Cook and published a list of 21 now famous problems [26] which he showed where all different versions of the same problem meaning that finding a polynomial time algorithm for one of them will yield a polynomial time algorithm for all of them. All of the problems was in this way shown to belong to the same complexity class namely the NP-complete class. This paper laid the foundation of the theory of NP-complete problems and

the problems on the list has become standard tools in determining the complexity of new computational problems. Since then many other problems has been shown to belong to the NPC class [1].

### Average case complexity and random ensembles

Even though NP-complete problems are hard in the worst case some instances can still be easy to solve. In the quest for a more detailed understanding of computational complexity Leonid Levin in 1986 introduced *average case* complexity [33] as a complementary complexity classification. In particular real world applications such as cryptography, one wants to be sure that almost all generated problems are hard, not only that some problems are. Hence the measure of a problems average case complexity can distinguish between different problems in the same worst case complexity class. One difference however between average and worst case complexity is that while worst case analysis covers *all* instances of a problem we need in the average case analysis to define what we mean by an *average instance*. What does a distribution of all possible instances of a problem look like? and how to calculate its average? Average case complexity only becomes meaningful in the presence of a probability function over a well defined *ensemble* of instances.

In the case of SAT one can generate distributions of instances in different ways by specifying ranges for parameters such as the number of literals in each clause, the number of clauses in the formula, the number of occurrences of a certain variable etc. The problem ensemble studied in this thesis is the set of randomly generated K-SAT instances generated by a probability distribution depending on the three variables  $N$ ,  $M$  and  $K$ .  $N$  specifies the number of variables,  $M$  the number of clauses (constraints) and  $K$  the number of literals in each clause. To generate an instance from this distribution one does the following. For each of the  $M$  clause,  $K$  out of the  $N$  variables are picked uniformly at random and turned into literals by negating the variables (with the NOT operator) with probability 0.5. The conjunction of all clauses is then a random instance from the ensemble. By varying the parameters different problem ensembles are generated with different average case behavior as we shall see below. The set of problem ensembles generated in this way is called *random K-SAT* and is the main problem class studied in this thesis. An important property of the instances from these ensembles is that the number of constrains that each variable belongs to will tend to a Poissonian probability distribution as  $N$  and  $M$  grows while their ratio remains fixed. The ratio  $M/N$  is denoted  $\alpha$  and describes the average level of constrainedness in the ensemble. A 'hard random K-SAT problem' will in this thesis mean an instance drawn from an ensemble with a high  $\alpha$ -value still below the SAT/UNSAT transition.

A motivation for the interest in randomly generated K-SAT instances is that by empirical studies [49] one has noted that algorithms that performs well on hard real world problems also performs well on hard random SAT problems which would presumably indicate that some important property of the original problem is still present.

```

DPLL(F,s)
  if F contains no clauses
    then s is a truth assignment; return TRUE
  if F contains an empty clause
    return FALSE
  fix value of unit clause literals
  fix value of literals which takes the same value in all clauses

  pick a literal x
  return DPLL(F,s(x=0)) OR DPLL(F,s(x=1))

```

Figure 1.2: The DPLL algorithm

## 1.2 SAT solving techniques

### Complete solvers

The most popular algorithm for solving satisfiability problems is DPLL [14] (Davis, Putnam, Logemann and Loveland). It uses a dynamic programming approach where it fixes the value of one literal, removes all clauses satisfied by this literal and remove the literal from all remaining clauses. This reduced formula is recursively sent back to the algorithm. If one ends up with an empty clauses then it cannot be satisfied and the algorithm returns 'FALSE'. If one ends up with a sub formula satisfying the remaining clauses the assignment is a solution and algorithm returns 'TRUE'. A clause containing only one literal automatically determines its value and exploiting this relation is called *unit propagation* and improves the performance. Another trick used in DPLL is to identify all literals which takes the same value in all clause. Such literals can also be fixed and removed. The procedure is outlined in figure 1.2

The original algorithm can be improved by selecting the literals to fix in a systematic way and applying other refined techniques to improve the backtracking procedure [13, 7]. In this thesis the complete solver used in paper number 4 and 5 is *relnat* [6] which is currently one of the most efficient solvers.

### Message passing

Since it is generally, even for relatively small problems, impossible to compute the cost value for each and every configuration (the power set scales exponentially with system size) one is, in many cases as mentioned above, restricted to use heuristic methods without solution guarantees. One should keep in mind that a complete solution in the context of optimization mean that if there is a solution, find one, otherwise report the problem as 'unsolvable'. Many heuristics however are only capable of reporting a problem instance as 'solvable' (by finding a solutions) but

not to with certainty find if it is unsolvable, that means that one needs to specify a time cut-off for the solver. Reaching the time cut-off does not ensure however that the problem instance is not solvable by any possible method which makes heuristic methods less attractive than proper algorithms from a complexity theoretical point of view. From a physics point of view on the other hand heuristic solvers can be seen as interesting dynamical objects in the energy (configuration) landscape and understanding the dynamic behavior of heuristic methods and how they relate to configuration space structure is a promising way to further understand the concept of computational hardness. Why are some heuristics more efficient on some problems than others? Two heuristic schemes related to the topics of this thesis that have been interpreted in a physics context are *local search* and *message passing*.

### Belief Propagation

A combinatorial optimization problem can be viewed as a bipartite graphical inference model. The graph modeling the constraints in the formula is called a *factor graph* where the nodes are bi-partitioned into *variables* and *constraints*. A variable node is connected to all the constraint nodes that contains the variable in the original problem and vice versa. A schematic picture of a factor graph for random 3-SAT is given in figure 1.6.

Message passing in the context of statistical inference was introduced by Pearl twenty years ago [45] and can be seen as a heuristic way to compute marginal distributions of node values in a graphical inference model. A graphical inference model describes how values of different nodes in a inference network are related, that is how the value of one variable effects the value of its neighbors. The most popular (and easy implementable) method is known as Belief Propagation (BP) and runs for many problems in linear time whereas the general problem of computing marginals in a general inference model is much much harder [55]. BP is an iterative schema where in each iteration each node calculates, based on the values received in the former iteration and its own value, what it *believes* the value of its neighbor is supposed to be. The procedure is continued until a fix point is reached and from the fix point beliefs one calculates the marginal probabilities. The method is proved to converge to the correct marginal distributions on all nodes as long as the underlying graph is *locally treelike* [56] meaning that the shortest path from a node to itself should scale at least as  $\log(N)$  where  $N$  is the number of nodes. This is not the case for many graphical models but empirical studies show surprisingly good results even for so called loopy networks. In random K-SAT and other random problems where each node has a finite number of network links the resulting graph will in the limit of large systems for sure locally look like a tree. This can be understood intuitively by imagine a broadcast message sent out from a certain node. The message will first be received by all neighboring nodes then their neighbors etc. The number of nodes reached in each step of the broadcast will scale with a multiplicative factor determined by the degree of the nodes. Since the nodes are connected at random,

the probability that a node that has already received the message gets it again (and therefore form a closed loop in the network) will be

$$p_{loop} = \frac{k^i}{N} \quad (1.1)$$

$$p_{loop} = 1 \Rightarrow k^i = N \Rightarrow i = \frac{\log(N)}{\log(k)} \propto \log(N) \quad (1.2)$$

Where  $i$  is the loop length. The graph will certainly have loops but they will not be shorter than  $\log(N)$  for large systems. This point is a crucial ingredient in applying many of the tools from statistical mechanics.

In the case of a factor graph one constructs a Belief Propagation message passing procedure where the variable sends messages to its constraint nodes about how probable it is that it will obey them and the constraint node sends message to all its variable nodes telling them the probability they will have to take a certain value in order for it to be satisfied. BP tries to estimate the marginal beliefs for each factor graph node which in turn can be used to calculate the probabilities that a node in the original combinatorial problem will take on a certain value (1 or 0). The fix point of belief propagation has been shown to correspond to extremal values of the Bethe free energy of the system defined by the Hamiltonian in equation (1.3)[56]. This means that once the BP equations find a fix point in the message passing procedure the values for the variables governed by this probability distribution will be an extremum to the original problem. This interesting result gives a clear physical interpretation of the output of an algorithmic program.

A supporting fact of the cluster transition, mentioned above, in the context of message passing is that the belief propagation equations do not seem to find a solution above that particular level of constraintness which could be interpreted as an indication of more than one set of solutions and therefore that multiple conflicting fix points are present in the problem.

### Survey propagation

Another great success for the physics community was the introduction of a novel message passing heuristic for solving very large highly constrained random satisfiability problems. The method is known as Survey Propagation (SP) [39, 38, 10] and is a set of BP equations run on the space of fix points to the original BP. The idea is to estimate the marginals of the fixpoints of BP and then iteratively decimate one variable at the time. The algorithmic implementation of SP [2] uses as a basis a version of Belief Propagation known as Warning Propagation in which one focuses on the variables forced to take a certain value 1 or 0. The analog in physics would be to look at the Bethe free energy at zero temperature. SP is then a way to calculate statistics over the fix points of Warning Propagation. The interpretation would then be that Survey Propagation is calculating statistics for the variables



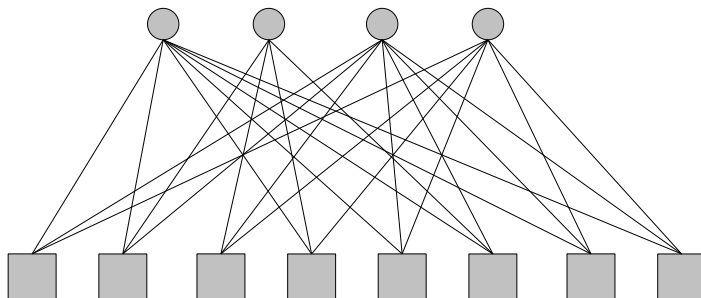


Figure 1.3: A cartoon picture of a factor graph representation of a random 3-SAT problem. The constraints are modeled as squares and the variables as dots.

over different solution clusters in the space of solutions. In the implementation of Survey Propagation one then uses decimation to gradually reduce the size of the problem. One runs SP until a fix point in the equations is found, one then picks out the top  $x$  variables with the highest bias towards either 1 or 0 and fix them to that value. Then iterate until no one of the variables left in the problem has a bias above a certain threshold. The physical interpretation would be that one gradually pin points a certain solution cluster. The main benefit so far has been that SP is able to solve very large random 3-SAT problems much more efficient than any currently known method. This has led to an increase in research interest both from physicist as well as from computer scientists on this specific method and message passing in general. Why SP, which as a solution method still is a heuristic with no solution guarantees, work so well is still somewhat a mystery and if its success has physical interpretations is either not clear yet. Current research in algorithm design is trying to clarify whether or not it is possible to generalize the method to problems with more inherent structure than random problems. A prerequisite for any statistical treatment of problem instances is that they are locally treelike meaning, as mentioned above, that the long range correlation between the values of the nodes are relatively weak, which in real world problems might not be the case.

### Stochastic local search

Local search methods are a class of heuristics for solving various types of optimization problems. The basic strategy used is to start from an arbitrary configuration and from there move to a *neighboring* configuration according to some rule. Neigh-

```
WalkSAT(F,p,t_max)
  X = random configuration
  DO WHILE t<t_max
    IF X satisfies F; exit

    pick any unsatisfied clause C
    IF variable in C can be flipped without increasing energy
      THEN flip one of them at random
    ELSE
      with probability p
        flip a random variable in C
      with probability (1-p)
        flip the best variable in C
    END IF

  LOOP
```

Figure 1.4: The WalkSAT algorithm

bor in this context means that only a small fraction of the variable assignments are altered (usually only one). In this thesis all examples will use a binary domain space and therefore two neighboring configurations differs in one single bit and movement in configuration space is done through consecutive bit flips. A *decision rule* determines where to move in each iteration of the search. The rule may be based on any type of information and even include random or partially random choices.

Traditionally SAT problems have been interesting from both practical and theoretical viewpoints. Many real-world problem can be transformed into SAT instances and has become one of the central model problems in complexity theory. Research for heuristic methods for solving SAT problems or *satisfiability testing* has therefore in past years been very extensive. I will not present a full review of the history of local search but rather give brief overview in order to put the performance of these methods in relation to solution space structure later on.

In 1982 Papadimitriou and Steiglitz introduced local search as an approximative method to handle NP-complete problems [44]. The approach was later subject to many numerical experiments on various local methods for SAT solving resulting in different improvements like GSAT[48], GWSAT[47], HSAT[21].

The overall behavior of a local search method is as mentioned above heavily dependent on the choice of neighbor selection rule. Imposing no bias in neighbor selection would result in a strict random walk on the set of possible configurations. Since the number of configurations in this set scales exponentially with number of

```

ASAT(F, p, t_max)
  WHILE t < t_max AND F is NOT satisfied
    flip a variable in an unsatisfied clause at random
    IF it increases the number of unsat clauses THEN
      flip it back with probability (1-p)
  LOOP

```

Figure 1.5: The ASAT heuristic. Optimal value for the noise parameter  $p$  is for 3-SAT 0.2

problem variables the chance of finding a satisfying assignment in reasonable time is very very small even for moderate number of constraints. An empirically better way to select neighbors is to monitor how many unsatisfied clauses one has before and after the move and base the decision on that value. Since one is in general interested in finding a satisfying assignment, biasing the search towards low values of unsatisfied clauses seems like a good idea.

The approach of introducing randomization in the variable selection, resulting in stochastic local search, has proved useful in many methods [50, 42]. Another very successful approach is to, in the update rule, only allow flips of variables in yet unsatisfied clauses [42]. This is by no means a trivial discovery since there might exist other variables, which when flipped, puts the system in a more favorable state than before. Such search strategies are called *focused*. The most efficient local methods for satisfiability seem to be the one that combines randomization and focusing. One example is the famous *RandomWalkSAT* algorithm in figure 1.2.

An important notion in local search is how *greedy* the search is. If the search only permit its downhill moves it will quickly find itself in a local optimum where none of the neighboring configurations has a lower number of unsatisfied clauses. It is therefore important to let the search once in while accept uphill moves which can be done by introducing a randomized decision. In most stochastic local search (SSL) heuristics there are then a number of parameters to be set such as noise level, probability of doing a uphill move etc. How these are set for optimal performance is basically done via trial and error. The idea that the only important parameter of a focused search is how greedy it is formed the basis for the ASAT heuristic introduced in paper 1. It is an extension of Papadimitriou's focusing algorithm [42] with the extension of rejecting uphill moves with a fixed probability  $(1 - p)$  (see figure 1.5).

Analysis of the performance of randomized methods on random K-SAT has been done [5, 51] giving valuable information on working ranges of local search.

### Physics interpretation

From a physicist's point of view a local search heuristic can be viewed as a dynamical object in the configuration space. Using the analogy between solution and ground

state one can interpret the difference in number of unsatisfied clauses as an energy gradient which can be used to guide the search. Favoring "downhill" flips would be analog of introducing a gravitational force in the system. This analog is often used in physical systems when one is interested in sampling typical configurations states. The methods used are called Monte-Carlo methods and will ideally give a representative sample of the equilibrium states present in the system. When looking for solutions on the other hand one is not interested in sampling typical states but rather to find the (very un-typical) ground states. All of the above mentioned local search methods interpreted as physical processes are therefore *non-equilibrium* processes (formally since they do not obey detailed balance). An optimal local search heuristic would then be one with its ground state equal to the set of solutions which it reaches as fast as possible without getting trapped in any local optimum.

## 1.3 A physics interpretation

### Phase transitions

Due mainly to its relation to deductive reasoning in the AI community many researchers were in the 90's interested in creating a way to generate random SAT instances which could be used as benchmark problems for new approximate solvers [48]. Some ensembles of random problems turned out to generate a high fraction of easy to solve instances meaning that one could trivially determine if the problem was satisfiable or not [19]. In 1992 however Mitchell et.al [40] showed empirically that when using the "fixed length model" (ever since synonymous with random K-SAT a usage followed in this thesis) the number of rounds needed for the Davis-Putnam (DP) algorithm [15] to classify a generated instance had a peak around the clause per variable ratio 4.25. The peak got sharper and sharper with increasing number of variables (keeping the ratio of constraints per variable fixed) and the authors therefore concluded that these problems would be the hardest ones. The reason for the statement was due to the fact that the clause to variable ratio at which the instances were satisfiable with 50% probability agreed also corresponded to a peak in the run times for the DP algorithm.

In a famous paper by Kirkpatrick and Selman in 1994 it was shown that when the clause to variable ratio is kept constant and the number of variables is increased, the fraction of unsatisfiable instances (smoothly increasing from 0 to 100%) obeys a finite size scaling approach similar to the behaviour of a physical phase transition. [30]. This means that the increment from 0% unsatisfied to 100% unsatisfied will eventually (with increasing system size) be abrupt. A small increase in the number of constraints per variable just at the critical point would then have a huge impact on the number of satisfiable instances. The underlying reason for the phase transition in the case of random 3-SAT was conjectured to be that the correlation between values of different variables would diverge exponentially in system size at the transition. This would imply that the variables would be very sensitive to the order in which the variables are set which would be the reason for the diverging run times. For low connectivity instances on the other hand the correlations would generally be small and the problem can be divided into a number of independent sub-problems, which can be solved independently, and is therefore easier to solve.

### Rigorous results on random 3-SAT

Since the discovery that random satisfiability problems displays critical behavior rigorous results have been shown in theoretical computer science community. The most relevant ones in the context of this thesis are summarized here. In 1999 Ehud Friedgut [20] proved that in 3-SAT, for any given value of  $n$ , there exists a critical density of constraints  $c^*(n)$  such that for any  $\epsilon > 0$  instances from distributions with  $c = c^* - \epsilon$  will asymptotically almost surely (a.a.s) be satisfiable while problems

with  $c = c^* + \epsilon$  will a.a.s be unsatisfiable. This shows that the transition is sharp but does not prove what the value of  $c^*$  will be in the limit  $n \rightarrow \infty$ .

Lower bound for the value of  $c^*$  exists for random 3-SAT. The best currently known value from 2003 is 3.52 [25, 23] which is proved by rigorously analyzing algorithmic performance. The best known upper bound from year 2000 is 4.506 [18] using a probabilistic proof technique such the first moment method.

For 2-SAT on the other hand the critical value is proved to be 1 [11, 22, 16].

The existence of clustering in K-SAT problems is proved for  $K \geq 8$  [34, 3] and also the existence of extensive barriers between clusters [27].

### State space structure

Methods such as *simulated annealing* [29] had early on showed that concepts and tools from physics can be used in computer science to design heuristic algorithms for optimization problems. In mapping the cost function (see above) of an optimization problem to an analogous energy function, the whole problem of finding an optimal configuration can be seen as finding a global energy minimum in an energy landscape. Many tools invented in the context of statistical description of physical system (statistical mechanics) are designed to study properties of global energy minimum or *ground states*.

The discovery of the SAT/UNSAT transition showed that not only ground state properties (number of solutions) but also critical behavior might be relevant in understanding computational complexity. This idea became the starting point for an intense (and still ongoing) activity in the theoretical physics community, with the overall goal to contribute to the understanding of fundamental processes in complexity theory. In order to see how a combinatorial optimization problem can be mapped onto a physical model one needs to understand a few important concepts.

A discrete energy model in physics is modeled through the *Hamiltonian function*  $H$ . This describes the constraints and therefore the cost function of the system. For 3-SAT the Hamiltonian function equals the number of unsatisfied clauses and looks like:

$$H = \sum_{m=1}^M \frac{1}{8} \prod_{i=1}^3 (1 - c_m^i x(m, i)) \quad (1.3)$$

The sum is over all  $M$  constraints and the  $c_m^i$  is the specification for the  $i^{\text{th}}$  literal of constraint  $m$ . The values of the  $N$  variables are stored in the  $x$  array. The term  $(1 - c_m^i x(m, i))$  is only non-zero if the variable does not meet its specification and thereby contributing to the sum (by a quantity  $2^3 = 8$  therefore the normalization).

### Equilibrium properties

One fundamental assumption made in statistical physics (which is not always valid in computer science models) is that each of the  $2^N$  possible configurations are

equally likely weighted only by the exponential of their cost value. From this assumption about the microscopic state of the system a number of macroscopic physical properties such as the entropy and free energy can be calculated. The connection between the microscopic and the macroscopic description of the system is made using Boltzmann's formula:

$$S = k_B \ln(W) \quad (1.4)$$

Where  $S$  is the entropy of the macroscopic state,  $W$  is the number of possible ways to order the microscopic system resulting in the macroscopic state and  $k_B$  is a physical constant. Since the Boltzmann constant  $k_B$  is only important to define an energy scale we can for simplicity set it to unity when dealing with non-physical computation models. The Boltzmann entropy equation can then be formulated, as a function of the probability distribution over the  $s$  microscopic states, in a so called information theoretic way (see [17] for more details) as:

$$S(p(i)) = - \sum_{i=1}^s p_i \ln(p_i) \quad (1.5)$$

If we now put the constraint on the system that the energy  $E = \langle H \rangle$  is conserved and that  $p$  is indeed a probability ( $\sum_i p_i = 1$ ) we can formulate the problem of finding the ground states as an optimization problem which can be solved by introducing two Lagrange multipliers to enforce the constraints. This is to say that the system will maximize the entropy given the constraints that energy is conserved.

$$g_1(p) = \sum_{i=1}^s p_i = 1 \quad (1.6)$$

$$g_2(p) = \sum_{i=1}^s E_i = C \quad (1.7)$$

$$\frac{\delta S}{\delta p_i} - \lambda_1 \frac{\delta g_1}{\delta p_i} - \lambda_2 \frac{\delta g_2}{\delta p_i} = 0 \Rightarrow \quad (1.8)$$

$$-1 - \ln(p_i) - \lambda_1 - \lambda_2 E_i = 0 \Rightarrow p_i = e^{-1 - \lambda_1 - \lambda_2 E_i} \quad (1.9)$$

Normalization gives:

$$p_i = \frac{p_i}{\sum_{j=1}^s p_j} = \frac{e^{-\lambda_2 E_i}}{Z} \quad (1.10)$$

The normalization constant  $Z$  is in physics called the partition function and the multiplier  $\lambda_2$  is usually denoted  $\beta$  and is in a physical system the inverse temperature but in combinatorial system can be viewed as inverse noise level. If  $\beta$  is very

large then the distribution will be sharply centered around the configurations that minimize  $H$ .

$$Z_0 = \sum_{j=1}^{s_0} e^{-\beta E_0} \quad (1.11)$$

Here the sum is taken over all the microscopic states that has the lowest energy  $E_0$  known as *ground states*. A combinatorial minimization problem is then equivalent to finding the ground state of the energy function  $H$ . The probability of picking a configuration with a certain non-minimal cost value (energy) from the set of all configurations is then proportional to:

$$p(\bar{x}) \propto e^{-\beta H(\bar{x})} \quad (1.12)$$

One of the most important contributions from the statistically physics community to the understanding of the onset of computational hardness in constraint satisfaction problems has been to predict another transition a so called *cluster transition* [9, 35, 36], at a slightly lower constraint density than the satisfiability threshold. A transition in the geometric distribution of the ground states (satisfying configurations). Below this transition the set of solutions were conjectured to be closely positioned in a large *cluster* whereas above the transition different areas of solutions would exist separated by extensive distance. Distance in this context is the Hamming distance or the number of bits differing between two configurations. The interpretation of the results in relation to computational hardness were that since a problem instance would contain many many more near optimal configurations (local ground states) than true solutions, heuristic solvers would quickly get lost and trapped in the enormous configuration space even for moderate system sizes.

The results were obtained through the quite elaborate *cavity method* [37] using advanced calculus and numerical methods such as *population dynamics* to solve complicated recursion relations. These methods are out of the scope for this thesis.

## 1.4 Solution space structure

The discovery of the cluster transition in the random K-SAT model introduced the idea that this critical behavior might be a good candidate for explaining the onset of computational hardness seen in both complete methods such as the DPLL algorithm as well as in the run times of local search. It was reasonable to believe that when increasing the number of variables in the problem a local method would have much higher chance (or risk) to find a local optimum than a global one and it was conjectured that these would be separated by extensive barriers (see e.g. [10]). Extensive in this context means scaling as some function of system size  $N$ . Theoretically, the cluster transition would separate a phase where more or less all available solutions would be located in a big cluster or homogeneously distributed across



the phase space. Above the transition on the other hand the solutions would be divided into exponentially many sub-clusters separated through non-zero Hamming distance. The location of the cluster transition is located at a ratio of constraints per variable value ( $\alpha$ ) about 3.9 [39] for 3-SAT. Since then the physics community has done extensive amount of work in categories different other transitions which might or might not be responsible for computational hardness. The description of the different solution space phases is described in [31] and an intuitive cartoon picture would look something like:

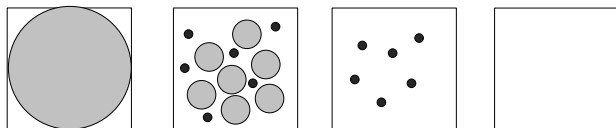


Figure 1.6: A intuitive picture of the different phases predicted by theoretical physicists. In the first phase all solutions are connected and has an extensive diameter. In the second phase the measure of solutions is dominated by an exponential number clusters with a much smaller diameter than the size of the system. After the so called *cluster* transition a typical solutions will with high probability belong to a cluster where at least one variable is frozen meaning it will take the same value in all solutions in that cluster. Finally above the SAT/UNSAT transition, all solutions are with high probability gone. The black dots are the frozen clusters.

In the first phase all solutions are connected in one cluster that covers more or less the whole state space. In the next state, the measure is dominated by a number of *typical* clusters meaning that if you draw a solution at random from the ensemble of all available solutions, you will with probability one when  $N \rightarrow \infty$ , find a solution that belongs to a typical cluster. Typical in this sense means that the number of clusters with a certain size weighted by its size is the maximum over all possible clusters. Therefore in the limit of large  $N$  the probability of finding something else than these typical solutions would be zero. The typical clusters would however be exponentially (in  $N$ ) numerous. Above the next transition the measure of solution density would be divided among a finite (non-exponential) number of clusters of the most typical size. The next thing to happen is that all solutions finally vanish. In recent time yet another transition is suggested in the context of graph coloring to be responsible for the onset of computational hardness. This transition has been labeled the *freezing* transition [52, 57]  $\alpha_f$  at which all clusters contains at least on variable which takes on the same value in all the containing solutions. The intuitive picture would be to imagine that a stochastic local search heuristic would have severe problems if it needs to pin point the exact value for an large number

of variables since its inherent randomness would make it keep fluctuating around some average configuration. It would experience a needle in a haystack kind of problem which in the limit of large systems would be intractable. If this is the case is still not known. No current stochastic local search approach can solve random 3-SAT all the way up to the SAT/UNSAT threshold but if this is due to freezing of variables or not is to early to say. Recent work on related models show that there is reason to believe that local search cannot find solutions when clusters are frozen [58]. A detailed discussion on the relation between stochastic local search and state space structure is found in the appended papers.

### Landscape experiment

In order for a local search method to be efficient on an optimization problem, the configuration space needs to have certain properties. As mentioned above, if there exists large barriers between local minimas the search will for sure get stuck. But yet another property of the space is important for a method dependent on the gradient namely that the space is not to flat. If almost all configurations have more or less the same energy then the search will get lost since there is no information to gain moving between states. In order to examine if the main reason for the onset of hardness in random 3-SAT is due to barriers (energy) or flatness (entropy) a simple experiment made. It is presented here since it is not contained in any of the papers.

Start by solving an instance with a local search solver. Then do the run again (with the same random seed) but this time monitor how far the search is from its final target. Doing this we get two plots for each experimental run. One describing how the number of unsatisfied clauses evolves with time and one which measures the distance from target with time. If the main reason for onset of hardness with increasing  $\alpha$  is energetic then one would see an increase in energy when the search 'escapes' the minima and the energy curve will fluctuate before finding a solutions. The results for three different  $\alpha$  values is shown in fig 1.9. For low values the energy decreases constantly as function of time and so does the distance to target. For high values of alpha on the other hand, the energy remains on a very low but non-zero value for a large amount of time whereas the distance to target is extensive. When a solutions is finally found it is done so in quite abruptly. An interpretation of the results suggests that the main reason for the onset of hardness is that the search cannot navigate in a space where there is no correlation between the states that has very low energy and the ones that are true solutions. In such spaces the search moves randomly and then finally finds a solution.

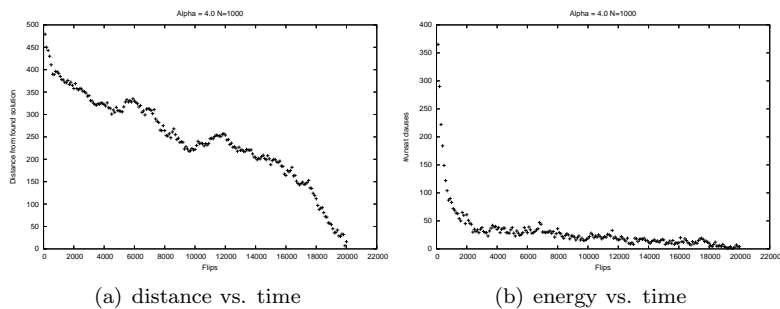


Figure 1.7:  $\alpha = 4$

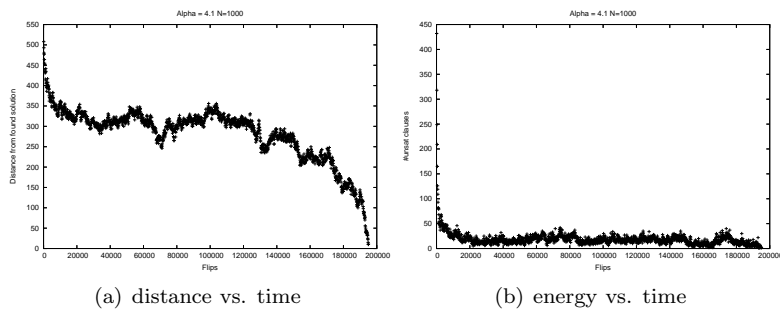


Figure 1.8:  $\alpha = 4.1$

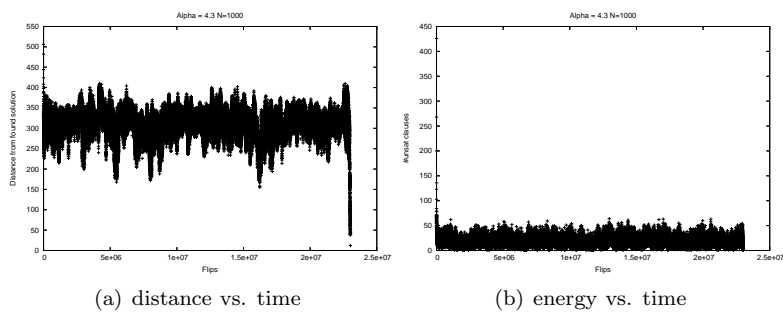
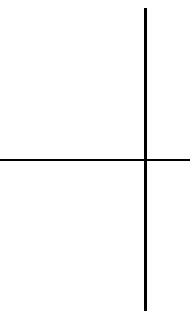
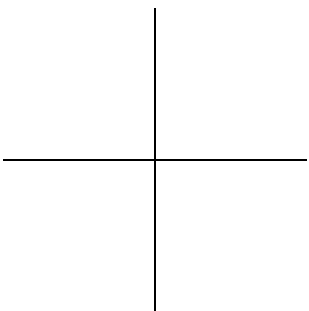
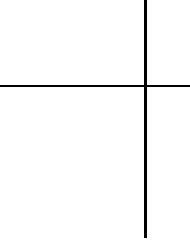
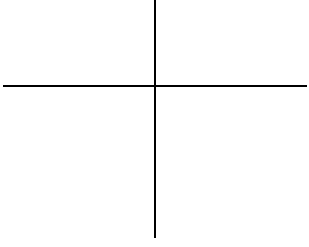


Figure 1.9:  $\alpha = 4.3$



## Chapter 2

# Papers and contributions

The initial motivation for the work done in this thesis came from the results from the theoretical physics community on a claimed relation between clustering of solutions and onset of hardness in constraint satisfaction problems [38, 39]. Our empirical experience showed quite early that problem instances way above the cluster transition was easily solvable even with such a simple heuristic as ASAT (see below). This triggered questions in the direction of how local search works on randomly generated problem instances and if it is possible to use computer experiments (that is finite size systems) to verify or falsify predictions based on methods from statistical mechanics. The notion of cluster used in the physics language was also not clear and how it was related to actual problem instances was also not known.

The result of the work has turned into five papers in the cross-section between physics and computer science.

### 2.1 Paper 1

John Ardelius and Erik Aurell

**Behavior of heuristics on large and hard satisfiability problems**

Phys. Rev. E 74, 37702 (2006)

The first paper written together with my supervisor Erik Aurell regards the working range and properties of local search methods near the SAT/UNSAT transition. In the paper a simplified version of the FMS [46] (which in turn is a variation of WalkSAT) called ASAT is introduced. ASAT is flipping variables only in unsatisfied clauses and rejects unfavorable moves with probability  $(1 - p)$ . Such an extremely simple heuristic without any form of intelligence or sophistication turned out to be one of the fastest and most successful local search methods for random K-SAT. The main purpose of the method was not however to find a fast solver but rather to explore how simple one can make a heuristic that still is efficient. Another

purpose still left undone is to implement the heuristic in a Markov model which would enable a more theoretical approach to understanding the dynamic behavior of stochastic local search.

Another contribution of the paper was that three different time scales in the course of solution was found. One where the search trivially and very quickly satisfies a lot of clauses followed by a phase where, still in linear time, a number of clauses is satisfied per time unit on a much slower time scale where at last a step-like behavior takes over where the remaining clauses are resolved in discrete events a few at the time.

### Contributions

I came up with the ASAT heuristic, did all simulations and wrote the first draft of the paper.

## 2.2 Paper 2

John Ardelius, Erik Aurell and Supriya Krishnamurthy

### Clustering of solutions in hard satisfiability problems

Journal of Statistical Mechanics (2007) P10012

The focus of this paper was structure of solutions found on instances when sampled with the local search heuristic ASAT from paper 1. The main motivation behind the work was to see how the concepts of clusters that had been vastly discussed in the physics literature applied to real moderate sized problem instances.

### Chaining

The paper introduces a convenient and later reused technique of *chaining* a problem instance. This procedure is carried out as follows; first a problem instance is generated at a constraintness level well above the SAT/UNSAT transition for a given number of variables  $N$ . One then tries to solve the instance with a arbitrary solver (in this case ASAT) , $s$  number of times. If a certain time cutoff is reached the current constraintness level is labeled as *unsolvable*. If the instance is solvable within the cutoff on the other hand, the solutions are saved and when  $s$  solutions are one calculates the mutual  $N^2/2$  overlaps between the solutions configurations. One then removes a constraint at random and continues the procedure until a minimum cutoff number in constrains is reached. The expected outcome would then be that (if the conjectures from the physics community holds) for low  $\alpha$  one would get many different values of the overlap (corresponding to a large uniform cluster) whereas for high constraintness levels (above the cluster transition) one would find the solutions confined to a set of separated parts of configuration space. This turned out to work and the results are discussed in more detail in the appended paper.

## Clustering

Another method developed in the paper was a heuristic way to geometrically categorize shapes and sizes of clusters. Once the set of  $s$  solutions has been collected one starts to cover the set in  $N$  dimensional circles called *rings*. At a first stage the radius of the rings are set to  $N$  and are placed with its center on a solution picked at random from the set. If not all solutions are covered by this circle one places another one at a random non-covered solution. One then continues until all solutions are covered. The final number of rings is recorded and one starts the procedure over again with a smaller ring radius. The outcome of the experiment will then show the number of rings needed to cover the set as a function of the radius of the rings. If the solutions were homogeneously spaced this function would be linear, otherwise it would show in what way the solutions are distributed. This method turned out to give some interesting insights in how the space of solutions is structured.

This paper's two greatest contributions (in my own opinion) were first to show that there actually exists a cluster of solutions at least when the space is sampled with a stochastic local search. The second contribution was to show that the sampled distribution undergoes a phase transition at a point of constraintness per variable after which all found solutions have a very small Hamming distance.

## Contributions

I came up with the idea to look at the solution space sampled by heuristic methods to see if that subset of the true space would be fragmented. I invented the chaining method to enable a systematic way to see the dependence on system size as well as the clustering method used to estimate the solution density. I did all simulations, some of the literature search and wrote the paper together with Erik Aurell and Supriya Krishnamurthy.

## 2.3 Paper 3

Submitted

Paper 3 is done in collaboration with our colleagues from Finland namely Mikko Alava, Pekka Orponen, Petteri Kaski and Sakari Seitz together with our group with me, Erik Aurell and Supriya Krishnamurthy. It explores the concepts of energetic barriers and solution density in the configuration space of random  $K$ -SAT problems. A novel and very interesting heuristic due to Sakari Seitz is introduced in the paper which can solve random  $K$ -SAT without doing a single uphill move which says something about how possible barriers might look.

Another important contribution made in the paper is to numerically understand the notion of *x-satisfiability*. The notion introduced in theoretical physics means that some intra solution distances are forbidden for a given constraintness level in

random K-SAT problems. This means that it is possible to find solutions only either very close to or very far from another solutions indicating a cluster differentiation of the solution space. In the paper it is shown by numerical methods that for 4-SAT the conjectured point of the onset of x-satisfiability does not show in experiments. It is possible, starting from an arbitrary solutions, to find other solutions at continuous distances.

### Contributions

The vast part of the numerical work was done by the Finnish group and my own contribution consisted mainly in discussions around how to design an experiment to verify or falsify the x-satisfiability theory in various ways, some of which made it into the final paper. I also did the comparison between the heuristic ASAT and SP.

## 2.4 Paper 4

Submitted

In this paper I am exploring the concept of sampling in randomly generated formulas. Previous work [54] had shown that certain local search methods would be good in sampling uniformly over the set of solutions in random problems and this work continues in the same spirit. The motivation for the work was to study the notion of *attraction basins* around sets of solutions. The idea was that since local search works well above the conjectured clustering transition, some clusters might be attractive in some way meaning that the configuration landscape around it would be flat or have some other properties such that finding solutions by local search would be easy. What I found was that when sampling the space with ASAT the sampling distribution over the set of clusters was not constant with respect to the level of the noise parameter. For high levels of noise the resulting sample would be more or less uniform over the set of solutions whereas for low values it became more and more uniform over the set of clusters. The intuitive interpretation would then be that any cluster is as hard to find as anyone else for a local search. The important properties is not only the number of solutions in the cluster but also the properties of the surrounding configuration space. A cluster with few solutions can still be surrounded by many many configurations with a small number of unsatisfied clauses which would make them just as easy for a local method to find.

Another construction introduced in the paper is a way to optimize the value for the noise parameter based on a physics inspired experiment called *simulated heating*. The procedure starts by trapping the search in a local minima by setting the noise level unreasonable low (or zero) until the search is stuck. During the whole procedure the number of unsatisfied clauses at any instant is recorded and eventually plotted as function of time. One then increases the value slightly and lets the



program run a number of flips much larger than the size of the system. This would be the analog to equilibrating a physical system. After that the noise level is again set to zero and the search is stuck. Then one sets the level a little bit higher than the last time and let the system equilibrate. This procedure is continued until the noise has reached a high enough level so that the number of unsatisfied clauses is relatively high. The result is very interesting in that the response in number of unsatisfied clauses to increase in noise level is not linear but displays a transition at a given value of noise. This value coincides with the empirically best known value obtained by trial and error. The outcome of the experiment is discussed in more detail in the paper.

## Contributions

I did all the work and wrote the paper on my own.

## 2.5 Paper 5

Submitted

This paper is written together with Lenka Zdeborova from Université Paris-Sud, Orsay. The paper is set out to investigate how well typical instances of small system sizes agrees with the predictions for infinite system models. In the physics literature (as stated above) the notion of a cluster, or a *state* is something counted by the Survey Propagation heuristic and is by no mean apparently equivalent to a cluster of solutions in a finite size satisfiability problem. By using a complete solver called Relsat [6] based on clause learning and refined backtracking procedures the complete set of solutions is generated for systems up to a few hundred variables and properties of each connected component in the space of solutions, such as size, whitening depth, number of frozen variables, etc. is examined. The result turned out to be in surprisingly good agreement with the theoretically calculated asymptotic values done using much more sophisticated methods. Based on this finding, we also used the chaining method mentioned above to pin point the onset of variable freezing in the clusters and thereby locating the so called freezing transition in random 3-SAT, a task that was not done earlier.

## Contributions

I did all simulations, came up with the idea to use chains to get the finite size scaling of the rigidity transition. Me and Lenka Zdeborova together came up with the way used to estimate the complexity function. I also contributed to many versions of the manuscript.

## 2.6 Conclusion

The most interesting parts of the work done in the past one and half years has mainly been two. First, quite a few people seem to have found the ASAT algorithm in one way or another and many of them find it useful. It has been used in related physics inspired work as well as for satisfiability testing as far as I know. From a research perspective the main contribution I feel is that the concept of clusters, cluster transition and various other abstract terms from theoretical physics has got analogous interpretations in real finite size systems. The work has shown that the notion of clusters are indeed an important concept in real problem instances and has also tried to shed light on its effect on stochastic local search methods. This work is however still far from finished and I think the interplay between physics and computer science has much to offer. The concepts of inherent randomness and problem ensembles are still quite new in computer science as compared to in physics and the community has, as I see it, much to gain from physics inspired problem solving, both actual techniques and mindset. From the computer science perspective one can conclude by saying that some of the conjectures made by the physicists regarding the implications of phase transitions to the performance of local search has turned out to be hard to verify on finite systems. On the other hand it is very good for the research area that theoretical predictions exists in order for others to verify or falsify.

The main questions still remains namely why the Survey Propagation method works so well as it does and why and how simple local search methods can solve these and other complicated combinatorial problems. Hopefully future work will address these and many more related questions.

# Bibliography

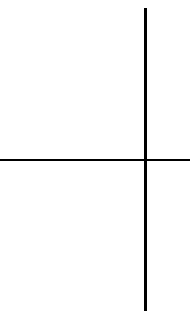
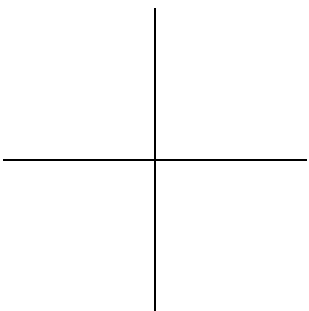
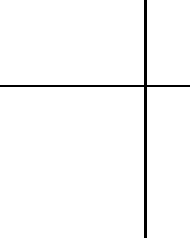
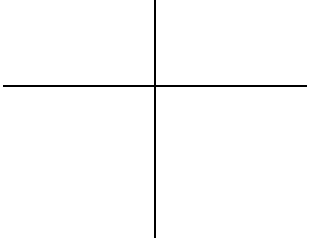
- [1] [http://en.wikipedia.org/wiki/List\\_of\\_NP-complete\\_problems](http://en.wikipedia.org/wiki/List_of_NP-complete_problems).
- [2] Code can be found at <http://users.ictp.it/zecchina/SP/>.
- [3] D. Achlioptas and F. Ricci-Tersenghi. On the solution-space geometry of random constraint satisfaction problems. *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 130–139, 2006.
- [4] G. Ausiello. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer, 1999.
- [5] W. Barthel, A.K. Hartmann, and M. Weigt. Solving satisfiability problems by fluctuations: The dynamics of stochastic local search algorithms. *Physical Review E*, 67(6):66104, 2003.
- [6] R. J. Bayardo and J. D. Pehousek. Counting models using connected components. In *Proc. 17th AAAI*, pages 157–162, Menlo Park, California, 2000. AAAI Press.
- [7] R.J. Bayardo Jr and R.C. Schrag. Using CSP look-back techniques to solve real-world SAT instances. *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, 203208, 1997.
- [8] R. Bejar and F. Manyá. Solving the round robin problem using propositional logic. *AAAI/IAAI*, pages 262–266, 2000.
- [9] G. Biroli, R. Monasson, and M. Weigt. A variational description of the ground state structure in random satisfiability problems. *Eur. Phys. J. B*, 14:551, 2000.
- [10] A. Braunstein, M. Mezard, and R. Zecchina. Survey propagation: an algorithm for satisfiability. *Random Structures and Algorithms*, 27(2):201–226, 2005.
- [11] V. Chvátal and B.A. Reed. *Mick Gets Some (the Odds are on His Side)*. Rheinische Friedrich-Wilhelms-Universität Bonn, 1992.
- [12] S.A. Cook. The complexity of theorem-proving procedures. *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, 1971.

- [13] J.M. Crawford and L.D. Auton. Experimental results on the crossover point in satisfiability problems. *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 21–27, 1993.
- [14] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, 1962.
- [15] M. Davis and H. Putnam. A Computing Procedure for Quantification Theory. *Journal of the ACM (JACM)*, 7(3):201–215, 1960.
- [16] W.F. de la Vega. On random 2-SAT. *Manuscript*, 1992.
- [17] K.A. Dill and S. Bromberg. *Molecular driving forces: statistical thermodynamics in chemistry and biology*. Garland Science, 2003.
- [18] O. Dubois, Y. Boufkhad, and J. Mandler. Typical random 3-SAT formulae and the satisfiability threshold. *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 126–127, 2000.
- [19] 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.
- [20] E. Friedgut. Sharp thresholds of graph properties, and the  $k$ -sat problem. *J. Amer. Math. Soc.*, 12, 1999.
- [21] I. Gent, T. Walsh, University of Edinburgh, and Dept. of Artificial Intelligence. *Towards an Understanding of Hill-climbing Procedures for SAT*. University of Edinburgh, Dept. of Artificial Intelligence, 1993.
- [22] A. Goerdt. A Threshold for Unsatisfiability. *Journal of Computer and System Sciences*, 53(3):469–486, 1996.
- [23] M.T. Hajiaghayi and G.B. Sorkin. The Satisfiability Threshold of Random 3-SAT Is at Least 3.52. *Arxiv preprint math.CO/0310193*, 2003.
- [24] H.H. Hoos. *Stochastic Local Search-Methods, Models, Applications*. IOS Press, 1999.
- [25] A.C. Kaporis, L.M. Kirousis, and E. Lalas. Selecting Complementary Pairs of Literals. *Electronic Notes in Discrete Mathematics*, 16:47–70, 2003.
- [26] R.M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, 43:85–103, 1972.
- [27] Petteri Kaski. Barriers and local minima in energy landscapes of stochastic local search. *Arxiv preprint cs.CC/0611103*, 2006.

- [28] H. Kautz and B. Selman. Pushing the envelope: Planning, propositional logic, and stochastic search. *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 11941201, 1996.
- [29] S. Kirkpatrick, CD Gelati Jr, and MP Vecchi. Optimization by Simulated Annealing. *Biology and Computation: A Physicist's Choice*, 1994.
- [30] Scott Kirkpatrick and Bart Selman. Critical behavior in the satisfiability of random boolean expressions. *Science*, 264:1297–1301, 1994.
- [31] Florent Krzakala and Lenka Zdeborova. Phase transitions and computational difficulty in random constraint satisfaction problems. *Proceedings of the International Workshop on Statistical-Mechanical Informatics, Kyoto*, 2007.
- [32] L.A. Levin. Universal search problems. *Problemy Peredachi Informatsii*, 9(3):265–266, 1973.
- [33] L.A. Levin. Average Case Complete Problems. *SIAM Journal on Computing*, 15:285, 1986.
- [34] M. Mézard, T. Mora, and R. Zecchina. Clustering of Solutions in the Random Satisfiability Problem. *Physical Review Letters*, 94(19):197205, 2005.
- [35] M. Mézard and G. Parisi. The bethe lattice spin glass revisited. *Eur. Phys. J. B*, 20:217, 2001.
- [36] M. Mézard and G. Parisi. The cavity method at zero temperature. *J. Stat. Phys.*, 111:1–34, 2003.
- [37] M. Mezard, G. Parisi, and M.A. Virasoro. *Spin glass theory and beyond*. World Scientific Teaneck, NJ, USA, 1987.
- [38] M. Mezard, G. Parisi, and R. Zecchina. Analytic and Algorithmic Solution of Random Satisfiability Problems. *Science*, 297(5582):812–815, 2002.
- [39] M. Mézard and R. Zecchina. Random K-satisfiability problem: From an analytic solution to an efficient algorithm. *Physical Review E*, 66(5):56126, 2002.
- [40] D. Mitchell, B. Selman, and H. Levesque. Hard and easy distributions of SAT problems. *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 459–465, 1992.
- [41] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky. Determining computational complexity from characteristic phase transitions'. *Nature*, 400(6740):133–137, 1999.
- [42] CH Papadimitriou. On selecting a satisfying truth assignment. *Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on*, pages 163–169, 1991.

- [43] C.H. Papadimitriou. *Computational complexity*. Addison-Wesley Reading, Mass, 1994.
- [44] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, 1982.
- [45] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [46] S. Seitz, M. Alava, and P. Orponen. Focused local search for random 3-satisfiability. *Journal of Statistical Mechanics: Theory and Experiment*, 2005:P06006, 2005.
- [47] B. Selman, H.A. Kautz, and B. Cohen. Noise strategies for improving local search. *Proc. AAAI*, 94:337–343, 1994.
- [48] B. Selman, H. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 440–446, 1992.
- [49] B. Selman, D.G. Mitchell, and H.J. Levesque. Generating hard satisfiability problems. *Artificial Intelligence*, 81(1-2):17–29, 1996.
- [50] Bart Selman, Henry Kautz, and Bram Cohen. Local search strategies for satisfiability testing. In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 26, 1996.
- [51] G. Semerjian and R. Monasson. A Study of Pure Random Walk on Random Satisfiability Problems with "Physical" Methods. *Theory and Applications of Satisfiability Testing: 6th International Conference, Sat 2003, Santa Margherita Ligure, Italy, May 5-8 2003: Selected Revised Papers*, 2004.
- [52] Guilhem Semerjian. On the freezing of variables in random constraint satisfaction problems. *J. Stat. Phys.*, 130:251, 2008.
- [53] M.N. Velev and R.E. Bryant. Effective use of Boolean satisfiability procedures in the formal verification of superscalar and VLIW microprocessors. *Journal of Symbolic Computation*, 35(2):73–106, 2003.
- [54] W. Wei, J. Erenrich, and B. Selman. Towards efficient sampling: Exploiting random walk strategies. *Proc. AAAI*, 4:670–676, 2004.
- [55] S. K. Michael Wong, Dan Wu, and Yiyu Yao. Critical remarks on the computational complexity in probabilistic inference. In *RSFDGrC*, pages 676–681, 2003.
- [56] J.S. Yedidia, W.T. Freeman, MA Cambridge, and Y. Weiss. Understanding Belief Propagation and Its Generalizations. *Exploring Artificial Intelligence in the New Millennium*, 2003.

- [57] L. Zdeborová and F. Krzakala. Phase transitions in the coloring of random graphs. *Phys. Rev. E*, 76:031131, 2007.
- [58] Lenka Zdeborová and Marc Mézard. Hard constraint satisfaction problems. *CoRR*, abs/0803.2955, 2008.





Part I  
Papers

