Generating Hard but Solvable SAT Formulas

T-79.7003 Research Course in Theoretical Computer Science

André Schumacher

October 18, 2007

1 Introduction

The 3-SAT problem is one of the well-known NP-hard problems that have been extensively studied over the years. Its generalisation, the k-SAT problem, is NP-complete for $k > 2$. In addition to being interesting from a theoretical viewpoint, 3-SAT also plays a role in industrial applications, such as boolean circuit verification. It is therefore of special interest to develop efficient solvers for reasonable problem sizes.

Although 3-SAT is NP-hard in general and all known algorithms for solving it require an exponential runtime in the worst case, it was recently observed that the performance of current state-of-the-art solvers over an ensemble of problem instances (determined by the range of different parameters, such as the ratio of clauses to variables) does not behave uniformly, but changes suddenly at critical parameter values. This sudden change in *typical* problem difficulty behaves similarly to an effect that is known from physics as *phase transition*. In order to understand changes in runtime performance that are connected to the structure of the underlying problem, models from physics provided useful information.

This seminar paper considers several recent publications that consider the application of spin-glass models to the generation of hard instances for the 3-SAT problem. The foundation on a structural model such as spin-glass models, which are well-known from physics, allows the analysis of the hardness of the problem from a statistical mechanics viewpoint.

1.1 3-SAT

Consider a given boolean formula F over a set $\{x_i | i = 1, \ldots, N\}$ of N boolean variables. The formula is assumed to be given in *Conjunctive Normal Form* (CNF), i.e. as a conjunction of disjunctions as described below. F consists of conjunction of a set $C = \{C_\mu | \mu = 1, \ldots, M\}$ of M logical clauses C_i. The ratio of M to N is denoted by α. Each clause is a disjunction of 3 *literals*, which are either direct or negated appearances of variables x_i. No variable appears twice in any clause. The task of the 3-SAT problem is to answer the question whether there exists an assignment A that defines mappings $x_i \mapsto \{\text{true}, \text{false}\} \equiv \{1, 0\}$ s.t. F evaluates to true.

1.2 Solvers

When considering solvers for boolean satisfiability problems, one has to distinguish between complete and incomplete solvers. Whereas complete solvers provide a complete coverage of the solution space, incomplete solvers typically maintain a current solution and aim at improving this particular solution during the run of the algorithm. For incomplete solvers, however, in general one can not guarantee that they will always find a satisfying truth assignment if the given formula is satisfiable.

As examples for complete solvers one should mention zChaff and Satz and for incomplete solvers WalkSAT, Record-to-Record Travel (RRT) and Survey Propagation (SP). As the range of problem instances to which incomplete solvers can be applied to is typically larger than the range of problems complete solvers are applicable to (due to the exponential size of the search space), it is of special interest to evaluate the performance of incomplete solvers on hard instances which are known to have satisfying truth assignments. In order to keep the effort for generating the instances low, it is not feasible to generate random instances and check them for satisfiability using complete solvers before using them as input to incomplete solvers.

2 Instance-generation algorithms

This section discusses several methods for generating random instances of the 3-SAT problem that are supposed to be hard for complete and particularly for incomplete solvers. The papers that have been considered for this seminar are the work done by Barthel et al. [1] and by Jia, Moore, and Selman [2]. In addition to the two papers, the work by Achlioptas, Jia, and Moore [3] is also mentioned within the context of a special case of the random generation of clauses proposed in [1].

2.1 Uniform random clause generation

As a possible candidate for generating random 3-SAT instances in CNF, one can consider the following procedure: First, one picks a random truth assignment A. Initially the set of clauses C is empty. In each iteration of the algorithm, one aims at adding a newly created random clause C_i to C until M clauses have been added. The clause C_i is comprised of three variables with indices $1 \leq i, j, k \leq N$ that are selected uniformly at random (without replacement). For each of the indices one flips a coin whether the corresponding variable will be included directly or negated in the clause. If the assignment A evaluates the clause C_i to true, C_i is added to C and the algorithm proceeds to the next iteration. Otherwise, the clause C_i is discarded and the previous iteration is repeated.

If one considers the methods for random clause generation described above, one notices that each variable appears on average in $4/7$ of all cases with the same value as in the assignment A and in $3/7$ of all cases with the opposite value. The algorithm therefore induces an imbalance towards the satisfying assignment.

One possibility to avoid the drift described above is to hide in addition to A also the complementary assignment $\bar{A}$, which means one also discards the clause that is not satisfied by $\bar{A}$. This approach was followed in [3]. However, this method introduces a strong dependence of the performance of incomplete solvers, such as WalkSAT, on the quality of the initial assignment. It is also the case that these problem instances are still solved by WalkSAT in time polynomial in the number of variables [1].

2.2 Generation of random 3-SAT instances based on spin-glass model for 3-SAT

2.2.1 Spin-glass model for 3-SAT

Spin-glass models originate from physics and can be used to study properties of materials such as magnetisation under external influences, such as magnetic fields or system temperature changes. They have been successfully applied to different optimisation problems that arise from practical applications but also to classical problems, such as 3-SAT, for example in the paper by Barthel et al. [1].

In general, a spin glass consists of a N Ising spins that can take two values, -1 or 1. The energy of a spin-glass *configuration* or *state* s is equal to the value of the *Hamiltonian* H for s.

$$H(s) = C - \sum_{i=1}^{N} H_i s_i - \sum_{i<j} T_{ij} s_i s_j - \sum_{i<j<k} J_{ijk} s_i s_j s_k \tag{1}$$

In this case we consider interactions between pairs and triplets of spins. The coefficients H, T and J in general can take any real values and the coefficient H is also called the *external field.* States that minimise the Hamiltonian are called *ground states.*

The task of deciding whether a given 3-SAT formula is satisfiable or not can be formulated as finding ground states for a particular spin glass whose spins corresponds to the boolean variables and whose Hamiltonian counts the number of unsatisfied clauses. If the Hamiltonian attains the value zero for a certain configuration, one can conclude that the corresponding truth assignment satisfies the given 3-SAT formula. Thus, the Hamiltonian H takes the form

$$H = \sum_{\mu=1}^{M} \frac{1}{8} \prod_{i=1}^{N} (1 - c_{\mu}^{(i)} s_i),$$

where the coefficients $c_\mu^{(i)}$ are defined as

$$c_\mu^{(i)} = \begin{cases} +1 & \text{if } x_i \text{ appears directly in } C_\mu. \\ -1 & \text{if } x_i \text{ appears negated in } C_\mu. \end{cases}$$

The boolean variables x_i are mapped to spins s_i by the mapping $s_i = (-1)^{1-x_i}$, which maps the value 0 for *false* to the spin value -1 and the value 1 for *true* to the spin value 1.

For the interaction coefficients and the constant C in (1) one obtains

$$C = \frac{M}{8} = \frac{\alpha}{8}N, \quad H_i = \frac{1}{8}\sum_{\mu=1}^{M} c_\mu^{(i)}, \tag{2}$$
$$T_{ij} = -\frac{1}{8}\sum_{\mu} c_\mu^{(i)} c_\mu^{(j)}, \quad J_{ijk} = \frac{1}{8}\sum_{\mu} c_\mu^{(i)} c_\mu^{(j)} c_\mu^{(k)}.$$

2.2.2 Clause generation process

Barthel et al. consider a generation process that utilises a probability distribution over the set of clauses satisfied by a given assignment A. Here, A is the assignment to be hidden in the random formula and without loss of generality it is assumed that A is the assignment $x_i = 1\ \forall i$. Note, however, that the hidden assignment is arbitrary as one can define variables $y_i = \bar{x}_i \oplus a_i$, where the a_i are random boolean values. In this case a satisfying assignment is the one that matches exactly the value of the random vector a, as the formula is satisfied if $y_i = 1\ \forall i$, which corresponds to $x_i = a_i\ \forall i$.

All clauses with the same number of negated variables (clause type) appear with equal probability by definition. The basic algorithm for constructing the problem instance is very similar to the method described in Section 2.1. In each iteration of the algorithm, one aims at adding a newly created random clause C_i to C until M clauses have been added. The clause C_i is comprised of three variables with indices $1 \le i, j, k \le N$ that are selected uniformly at random (without replacement). However, instead of flipping a coin whether or not the variable will appear negated in the clause, each clause is generated according to the probabilities shown in Table 1. Note that the only clause that is not satisfied by the assignment A (the clause in which all variables appear negated) has probability zero of being constructed.

The generation probabilities p_i for each of the possible clause types have to fulfil the constraints

$$0 \le p_i \le 1\ \forall i \in \{1,2,3\}, \quad p_0 + 3p_1 + 3p_2 = 1.$$

The range of possible p_i can be analysed within the context of the spin-glass model described above. For the interaction coefficients in (2) one obtains the following statistical averages.

$$\overline{H_i} = \frac{3\alpha}{8}\left(p_0 + p_1 - p_2\right), \quad \overline{T_{ij}} = \frac{3\alpha}{4N}(-p_0+p_1+p_2), \quad \overline{J_{ijk}} = \frac{3\alpha}{4N^2}(p_0-3p_1+3p_2) \tag{3}$$

Clauses	Probability
$(x_i \vee x_j \vee x_k)$	p_0
$(x_i \vee x_j \vee \bar{x}_k)$, $(x_i \vee \bar{x}_j \vee x_k)$, $(\bar{x}_i \vee x_j \vee x_k)$	p_1
$(x_i \vee \bar{x}_j \vee \bar{x}_k)$, $(\bar{x}_i \vee x_j \vee \bar{x}_k)$, $(\bar{x}_i \vee \bar{x}_j \vee x_k)$	p_2

Table 1: Generation probabilities for different types of clauses. Note that the clause $(\bar{x}_i \vee \bar{x}_j \vee \bar{x}_k)$ is not satisfied by the assignment and therefore discarded.

2.2.3 Observations for different generation probability ranges

The first interesting observation that the spin-glass model allows, is the case of equal generation probabilities for all clauses that are satisfied by the assignment A, as it was described in Section 2.1. In this case one chooses $p_0 = p_1 = p_2 = 1/7$. Based on the spin-glass model discussed previously, one observes that the average value for the external field $\overline{H_i}$ is non-zero. More precisely, one obtains that $\overline{H_i} = 3\alpha/56$, so that incomplete methods, such as WalkSAT, are guided by the field towards the hidden assignment. Therefore, in order to create instances which do not reveal any information about the satisfying assignment, it is beneficial to add the constraint

$$p_0 + p_1 - p_2 = 0,$$

which forces the external field to be zero. Resulting from the previous restriction, one obtains the following inequality and equalities that the p_i have to fulfil.

$$0 \leq p_0 \leq \frac{1}{4}, \quad p_1 = \frac{1 - 4p_0}{6}, \quad p_2 = \frac{1 + 2p_0}{6} \tag{4}$$

The second case that can be considered interesting is the situation in which the complementary assignment $\overline{A}$ is hidden in the formula in addition to the satisfying assignment A, which was also described in Section 2.1. This generation method can be modelled by choosing $p_0 = 0$ and $p_1 = p_2 = 1/6$. For this particular choice of generation probabilities the interaction coefficient $\overline{J_{ijk}}$ vanishes. One observes that the backbone (the value of variables that is the same in any satisfying assignment) is formed only at relatively high values of α and only then starts to appear continuously. Further analysis in [1] reveals that at the point of appearance of the backbone, the system has already entered a ferromagnetic phase with increased magnetisation of spins on average, which makes the backbone in some sense easier to spot for local algorithms. Indeed, WalkSAT shows good performance when run on formulas that were obtained in this region of the problem instance space, as one can observe from Fig. 1. Recall that finding the backbone is one of the major obstacles for incomplete algorithms, as choosing the value of only one variable permanently different from the value in the backbone renders the search for a satisfying assignment unsuccessful.

When choosing $p_0 \neq 0$, the hardness of the resulting instances changes dramatically. As a special case one can consider choosing $p_0 = p_2 = 1/4$ and $p_1 = 0$.

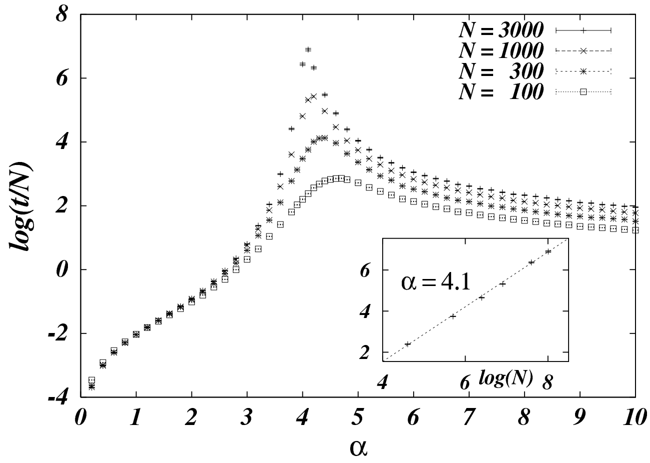

Figure 1: Performance results for runs of WalkSAT on instances generated with $p_0 = 0$ and $p_1 = p_2 = 1/6$; picture taken from [1].

It turns out that the resulting instances are from the category of random satisfiable 3-XOR-SAT formulas, which means that the clauses can be rewritten to the form $x_i \oplus x_j \oplus x_k$. It has been shown previously that this type of problems can be solved efficiently with a polynomial time algorithm similar to the Gauss elimination method for solving linear equations. If one considers random instances from this domain, however, incomplete methods such as WalkSAT show an exponential execution time on average. A possible reason for this behaviour can be found in the discontinuous formation of the backbone which occurs already at lower values of α, where the magnetisation is low as well. A further investigation of the hardness of 3-XOR-SAT formulas is discussed in the following section. See Fig. 2 for the results that were obtained for this choice of parameters.

The range $(0, 1/4)$ for p_0, excluding the boundary values, results in very different instances. For $0.077 \lesssim p_0 < 1/4$ the results presented in [1] show that the formation of the backbone happens discontinuously. This effect is a potential reason for the observation that incomplete methods, such as WalkSAT, have difficulties in this region to find satisfying assignments and typically require an exponential running time.

2.3 Generation of instances based on triangular lattice spin glass

2.3.1 Spin-glass model for 3-XOR-SAT

Jia, Moore, and Selman [2] consider an Ising spin-glass model on a triangular lattice structure with nearest-neighbour interaction and short loops. From physics is already known that this kind of model shows *glassy* behaviour, in terms of its

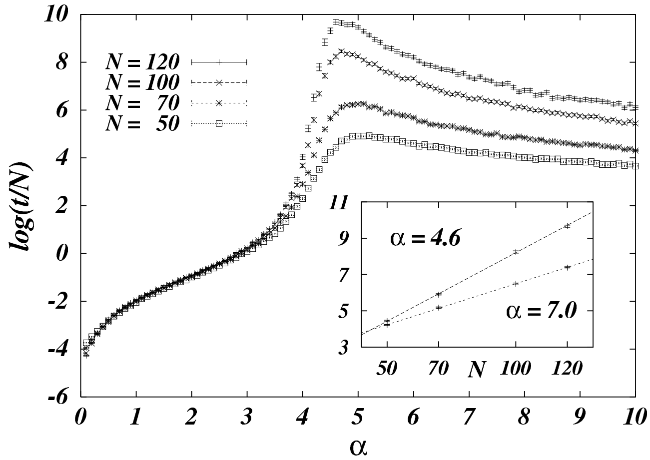

Figure 2: Performance results for runs of WalkSAT on instances generated with $p_0 = p_2 = 1/4$ and $p_1 = 0$; picture taken from [1].

Hamiltonian having a large number of local minima with high-energy barriers between them. However, due to the regular structure of the model, one can say a lot more bout the hardness of 3-SAT instances that result from it.

The connectivity structure of the spin-glass model discussed in this section is a $L \times L$ rhombus with periodic boundary conditions and its Hamiltonian H is defined as

$$H(s) = \frac{1}{2} \sum_{i,j=0}^{L-1} s_{i,j} \cdot s_{i,j+1 \bmod L} \cdot s_{i+1 \bmod L,j}, \tag{5}$$

where the summation runs over all downward pointing triangles in the lattice. Figure 3 depicts a triangular lattice with side length $L = 2^2$ and indicates the nearest-neighbour interactions between nodes on a downward pointing triangle. The periodic boundary conditions are omitted from the picture for the sake of clarity.

The Hamiltonian (5) can be rewritten using boolean variables $x_{i,j}$ that are obtained from the spins by the mapping $x_{i,j} = \frac{1}{2}(s_{i,j} + 1)$. Up to a constant one obtains

$$H = \sum_{i,j=0}^{L-1} ((x_{i,j} + x_{i,j+1 \bmod L} + x_{i+1 \bmod L,j}) \bmod 2). \tag{6}$$

Note that this representation corresponds to L^2 3-XOR-SAT clauses of the

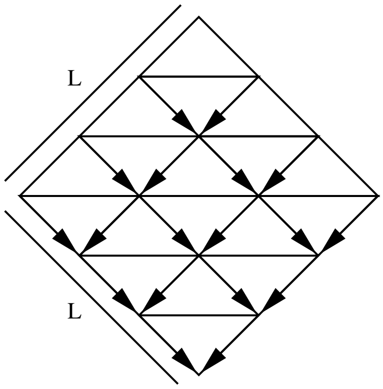

Figure 3: Triangular lattice with $L = 2^2$ and spin interactions; the arrows are indicating the triangles of the spins interacting with each other; periodic boundary conditions are omitted for the sake of clarity.

form

$$\begin{aligned}
&\overline{x_{i,j} \oplus x_{i,j+1 \bmod L} \oplus x_{i+1 \bmod L,j}} \\
&\equiv (\bar{x}_{i,j} \vee x_{i,j+1 \bmod L} \vee x_{i+1 \bmod L,j}) \wedge (x_{i,j} \vee \bar{x}_{i,j+1 \bmod L} \vee x_{i+1 \bmod L,j}) \wedge \\
&\quad (x_{i,j} \vee x_{i,j+1 \bmod L} \vee \bar{x}_{i+1 \bmod L,j}) \wedge (\bar{x}_{i,j} \vee \bar{x}_{i,j+1 \bmod L} \vee \bar{x}_{i+1 \bmod L,j}).
\end{aligned}$$

The resulting representation is a 3-SAT formula with L^2 variables and $4L^2$ clauses and the Hamiltonian H counts the number of unsatisfied clauses. The assignment $x_{i,j} = 0 \; \forall i, j$ is always satisfying and one can show that it is indeed the unique satisfying assignment if $L = 2^k$ [4]. Note that, as discussed previously, the hidden assignment is arbitrary if one defines variables $y_{i,j} = x_{i,j} \oplus a_{i,j}$, where the $a_{i,j}$ are random boolean values. The problem of finding a satisfying assignment is solvable in polynomial time by Gauss elimination modulo 2 but for local solvers the problem of finding a satisfying assignment is challenging.

2.3.2 Hardness of formulas for SAT solvers

Note that a satisfying assignment which corresponds to a ground state of the Hamiltonian (5) implies

$$x_{i,j+1 \bmod L} = x_{i,j} \oplus x_{i+1 \bmod L,j}, \tag{7}$$

so that in fact the boolean values are given by Pascal's triangle modulo 2. In order to determine the hardness of the resulting boolean formulas, one can consider an assignment A' that unsatisfies exactly one clause. Define the variables $d_i := x_{i,j} \oplus x_{i,j+1 \bmod L} \oplus x_{i+1 \bmod L,j}$, which we call *defects*. Figure 4 depicts such a configuration. One should note that in order to satisfy the remaining

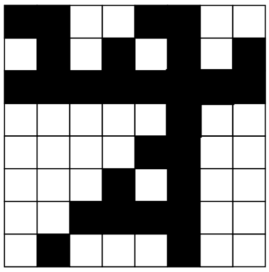

Figure 4: Spin glass configuration with a single defect corresponding to an unsatisfied clause; white cells corresponds to 0 values, black cells correspond to 1 values.

clauses, a certain number of variables have to be set to the value 1, as indicated in the figure. It can be shown that the number of ones in the resulting assignment is $L^{\log_2 3}$, which is also the Hamming distance to the satisfying assignment, $x_{i,j} = 0 \; \forall i, j$ [4][1].

Furthermore, it is possible to take linear combinations of single-defect assignments and form assignments with an arbitrary number of defects. One can show that if these defects form an independent set on the triangular lattice, then the corresponding spin-glass state is a local minimum of the Hamiltonian. Therefore, the number of local minima scales as the number of independent sets on the triangular lattice, which is $O(\kappa^{L^2})$, where $\kappa \approx 1.395$ (*hard hexagon constant*) [2].

Considering the energy barrier between local minima and the unique satisfying assignment, analytical results show that in order to escape a local minimum, once has to introduce $O(\log_2 L)$ additional defects. To summarise, this means that there are a large number of local minima which all are at a large Hamming distance from the satisfying assignment and which are separated by a large energy barrier from the satisfying assignment. These properties are responsible for the glassiness of the spin-glass model and therefore for the hardness of the resulting 3-SAT instances.

3 Conclusions

The problem of generating 3-SAT instances that have a previously known satisfying truth assignment is important for the evaluation of solvers, in particular incomplete solvers. Generating instances purely at random and disregarding the unsatisfiable ones is typically not feasible for practical reasons, i.e. it would in general lead to an exponential running time of the algorithm for constructing problem instances.

Considering the hardness of instances that are generated by various schemes,

[1] Curiously enough, this number does not match the number of ones in Fig. 4; there seems to be a factor of 3 missing.

one can observe that not all “random” generation schemes result in hard formulas. In fact, although the 3-SAT is NP-complete and hard instances exist in general, there are important subclasses of formulas, such as the XOR-SAT formulas, that are solvable in polynomial time. These formulas still pose a challenge for solvers. One of the models that are discussed in this seminar paper considers this interesting subclass.

The parameterised generation of instances can reveal interesting aspects of problem structure. One of the two spin-glass models that was discussed in this seminar paper employs a parametrised random generation of clauses which shows such a shift in complexity for the resulting input instances.

References

[1] Barthel, W., Hartmann, A.K., Leone, M., Ricci-Tersenghi, F., Weigt, M., Zecchina, R.: Hiding solutions in random satisfiability problems: A statistical mechanics approach. Phys. Rev. Lett. **88**(18) (2002) 188701

[2] Jia, H., Moore, C., Selman, B.: From spin glasses to hard satisfiable formulas. In: Theory and Applications of Satisfiability Testing. Volume 3542/2005., Springer Berlin / Heidelberg (2005) 199–210

[3] Achlioptas, D., Jia, H., Moore, C.: Hiding satisfying assignments: Two are better than one. J. Artif. Intell. Res. (JAIR) **24** (2005) 623–639

[4] Newman, M., Moore, C.: Glassy dynamics in an exactly solvable spin model. Phys. Rev. Lett. **60** (1999) 5068–5072