Phase Transitions in Generalized SAT-problems

Summary for workshop on Recent developments in phase transitions in optimization problems

by Juha Koivisto

1 Introduction

Imagine you organize a dinner party. You want your closest friends John, Alice and Tom to join. These three are not the most easiest people. They have restrictions, they want to enjoy your party, but they cannot unless there is someone they like or there is not someone they dislike. For example John says: 'I come only if mr Black comes and mr Green does not'. The other two of your friends have similar conditions. You have a satisfiability (SAT) problem in your hands: Who else you can invite, if you want John, Alice and Tom to be there?

John has two conditions corresponding to a problem with 2 literals per clause, a 2-SAT problem. If Alice has 3 conditions (resp. 3-SAT), an interesting question raises about the hardness of the problem. When the problem with mixture of 2 and 3 clauses is solvable in polynomial time like 2-SAT and when in exponential time like 3-SAT. Is there a transition, in which point and what properties the transition has?

2 K-SAT

SAT-problems consist of boolean variables. Literals are varibles or their negations. Literals grouped together with OR's are clauses, which are connected together with AND's forming the formulae to satisfy. In the dinner party example John picked two quests (literals) from the group of other quests (variables) and negated one of them.

The dinner-party example is an easy problem if all 3 friends have only 2 literals in their clauses. The problem is then called 2-SAT, which can be solved in polynomial time. Therefore 2-SAT is a not an NP-complete (non-deterministic polynomial time) problem. If the number of literals is 3 or more the solution is not solvable in polynomial time with the algorithms known so far and the problem is NP-complete. The definition for NP-completeness requires that the problem is NP and all other NP-problems can be reduced to it. In order to solve all NP-complete problems efficiently[1] we need to solve only one efficiently.

Satisfiability problems have two interesting properties: Only one of the literals has to be true to make the clause true and only one clause has to be false to make the formulae false. This means that by increasing the number of literals in a clause, the formulae becomes easier to satisfy and by increasing the number of clauses, the formulae becomes harder to satisfy. The order parameter α

[1] in polynomial time, if possible

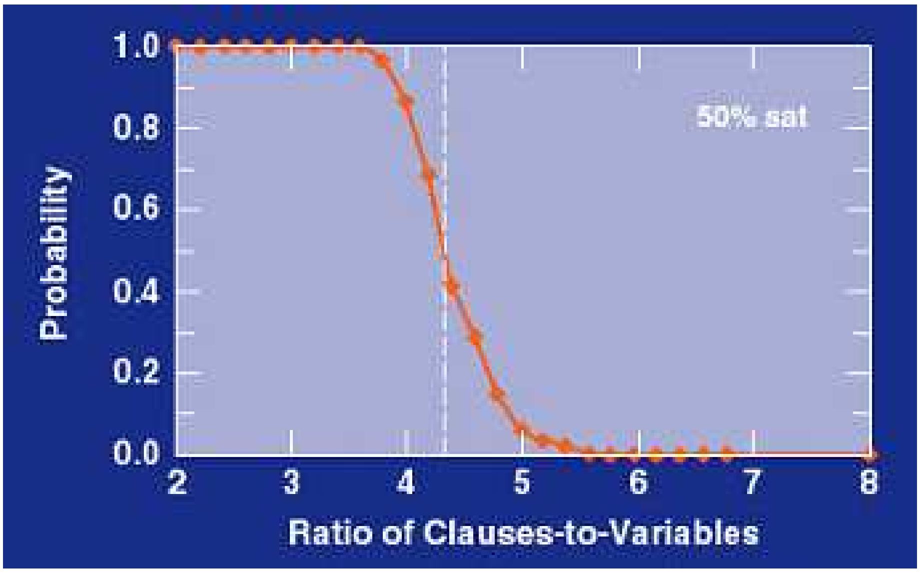


Figure 1: Fraction of satisfied formulae as a function of order parameter α in 3-SAT from [1].

is defined as number of clauses (M) per number of variables (N), $\alpha = M/N$. Figure 1 shows the transition from all satisfied to all unsatisfied as the number of clauses increase. The literals are randomly chosen from variables or their negations with equal probability. The vertical line shows the critical value, α_c, at the point where 50% of the clauses are satisfied. For 3-SAT $\alpha_c = 4.27$ [2] and for the 2-SAT $\alpha_c = 1$.

Monasson et al.[3] have shown earlier that the phase transition from all satisfied to all unsatisfied is continous with 2-SAT and discontinous with 3-SAT. The fully constrained variables are identified as backbone. These variables are either true or false for each satifiable clause. The fraction of fully constrained variables below and above α_c is continuous for 2-SAT and jumps from 0 to finite value for 3-SAT.

Further, Monasson et al. [4] investigated a formulae with both 2 and 3 clauses, 2+p-SAT, where p is the fraction of 3-clauses. They have shown that there exists a threshold $p_0 \sim 0.41$ which divides the region of $p = [2, 3]$ into continous ($p < p_0$) and discontinuous phase transition ($p > p_0$).

3 Algorithms

For determining whether the given formulae is satisfiable or not, a suitable search algorithm is to be used. 2-SAT can be solved in linear time, but all methods for 3-SAT have an exponential worst-case solving time. The simplest

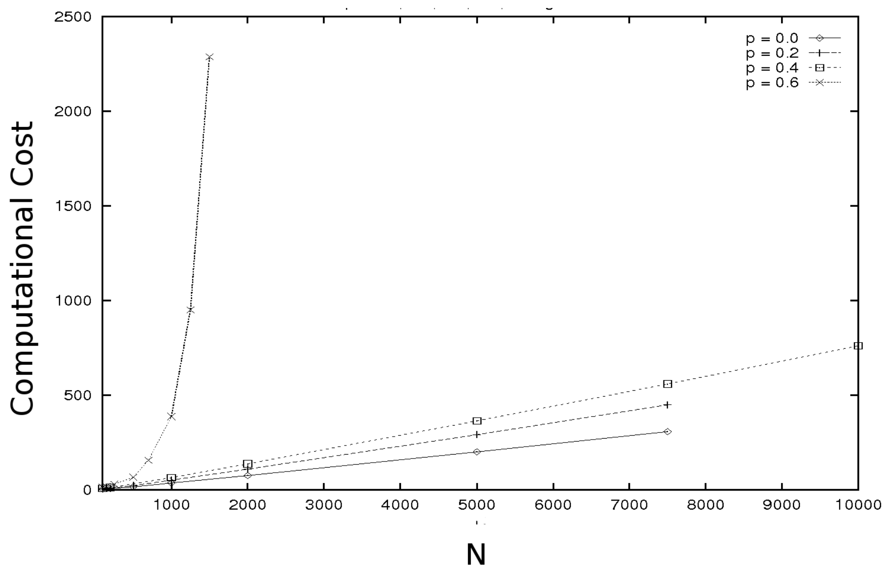


Figure 2: Computational cost as a function of the number of variables, N, for different values of p from [4].

algorithm is the branch & bound that does tree search and stops if the solution is found. If no solution is found, the problem is unsatisfiable. Monasson et al. used two differnt algortihms for experimental determination of $\alpha_c(p)$.

3.1 Tableau

A more advanced algorithm to B&B is the tableau algorithm [2]. It is based on brach & bound but has optimizations to cut off unsatisfiable branches of the search tree. The tableau is a model search algorithm meaning it searches for solution for given formulae.

The tableau algorithm evaluates literals (and their negations) similar to branch & bound. After the literal is evaluated follows the unit propagation phase where the formulae are simplified with the help of evaluated literals.

The computational cost for different values of p is measured using the tableau algorithm. The results are shown in Figure 2. The cost is linear for $p < p_0$ and exponential for $p > p_0$. For determining α_c the tableau algorithm was used only for p larger than p_0. The tableau algorithm is not so efficient in redundant search spaces (such as random 2-SAT) as the MODOC-algorithm.

3.2 MODOC

The MODOC algorithm does refutation[2] and model search in parallel [5]. In addition of finding a solution to the formulae it searches contradictions in the unsatisfiable phase. The key idea in MODOC is the pruning of autarkies, the self-sufficiencies.

An autarky is defined as a partial assignment[3] M that divides a set of CNF-clauses into two parts as

$$S = autsat(S, M) + autrem(S, M), \tag{1}$$

where autsat(S,M) is a set of clauses which can be satisfied by M and autrem(S,M) is a set of clauses with no common literals or their negations with M. As M is a partial assignment, autarkies can include autarkies.

The MODOC algorithm forms a hierarky of autarkies, which speeds up the refutation and model search in redundant search spaces. The MODOC algorithm works well for 2-SATs, while it prunes away identities e.g.

$$(a \vee b) \wedge (\bar{b} \vee c) = a \vee c, \tag{2}$$

where a, b and c are literals. These combinations are common close to the phase boundary $\alpha_c = 1$ since the number of literals in a formulae is twice the number of variables and only thing you need is a one variable and its negation to simplify the formulae.

The modoc algorithm excels with the redundant, structured search spaces such as $p < p_0$ but underperforms the tableau in unstructured cases such as $p > p_0$. In the latter case the autarky pruning takes more time than giving advantage.

4 The Phase Transition in 2+p SAT

The most interesting property of 2+p SAT problem is when it changes from an easy (linear computational cost) to a hard (exponential cost) problem. This happens near the phase boundary. This section is based on reference [4]

The system is assumed to have replica symmetry (RS) near $p = 0$, 2-SAT. A more sophisticated method called Replica Symmetry Breaking (RSB) should be used near $p = 1$, 3-SAT. The difference between RS and RSB is that RS approximates the cost function with an energy barrier with one simple valley, while in RSB approksimation there are many valleys.

A cost function of the form

$$E[C, S] \quad = \quad \sum_{i=1}^{(1-p)M} \delta_2(C_i, S) + \sum_{i=(1-p)M+1}^{M} \delta_3(C_i, S) \tag{3}$$

[2] search for contradiction

[3] a partial function from literals to {true,false}

$$\delta_K(C_i, S) \quad = \quad \begin{cases} 1 & \text{, clause } C_i \text{ is not satisfied by } S \\ 0 & \text{, otherwise} \end{cases} \tag{4}$$

is introduced, where C_i are the clauses, M is the number of clauses, S is the state, configuration, to be tested whether it satisfies the clauses. The above equation essentially tells the number of violated clauses. If the minimum of $E[C, S]$ respect to S is zero the formulae is satisfiable, else its unsatisfiable.

$$E[C] = \min_S E[C, S] \tag{5}$$

The α_c can be located by examining the behaviour of $<< E[C] >>$ the avarege over disorder of $E[C]$ in thermodynamic limit. $<< E[C] >>$ is zero in the satisfied and non-zero in the unsatisfied region.

Mathematically $<< E[C] >>$ can be calculated using a partition function, energy approximation and a replica trick. The partition function is the sum over all the states S

$$Z[C] = \sum_S \exp(-E[C, S]/T). \tag{6}$$

The energy of the system can be approximated as

$$<< E[C] >> \sim \; -T << \log Z[C] >> . \tag{7}$$

The above sum over all states is impossible to calculate for large systems. Therefore a replica trick is used to calculate the average of Z[C] by replicating n times the partition function and by taking the limit $n \to 0$

$$\lim_{n \to 0} \frac{Z^n - 1}{n} = \log Z. \tag{8}$$

The treatment above makes it possible to calculate the energy average $<< E >>$. There exists two solutions. The first solution has a continuous phase transition i.e. no jump in the backbone fraction at α_c for $p < p_0$. The other solution has a discontinuous phase transition from zero to a finite value at α_c for $p > p_0$. These two behaviours fix the parameters p_0 and $\alpha_c(p)$. An iterative approximation allows to obtain $p_0 \sim 0.41$.

Figure 3 depicts the α_c as the function of p solved from the replica theory (solid line) and from the random 2+p-SAT experiments solved by the tableau and MODOC -algorithms. The RS-scheme is believed to be exact below p_0, while above p_0 it gives an upper limit which can be seen in the Figure 3.

Figures 4 depicts the backbone fractions of the variables. For 2-SAT the fraction approaches zero at the phase boundary α_c for large N. For 3-SAT there is a jump from 0 to 0.45 in the backbone fraction at α_c. Notice that the formulae size N is less than 30 for 3-SAT.

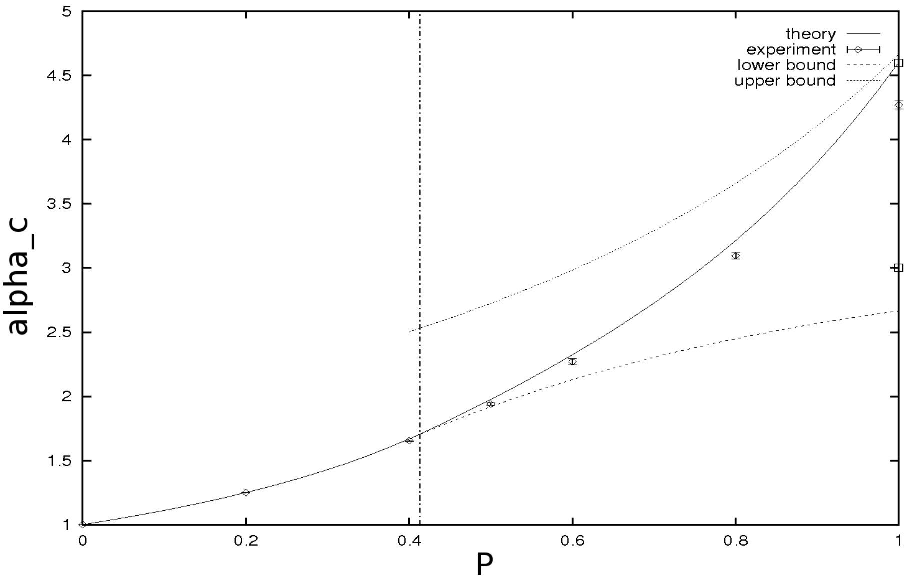
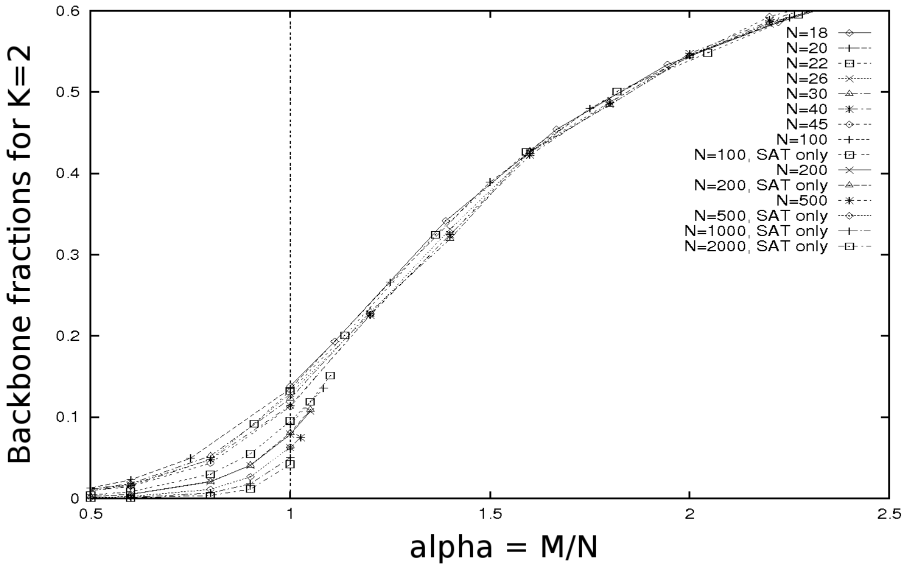


Figure 3: Phase diagram of satified and unsatisfied phases from [4].

5 Conclusions

The 2+p-SAT problems can be divided into two parts. When $p < p_0$ the problem has replica symmetry, a continuous phase transition and is computationally easy (linear solving time). When $p > p_0$ the replica symmetry is broken, the phase transition is discontinous and the problem is computationally hard (exponential solving time). The value of p_0 can be approximated to 0.41. The theoretical results were verified by experiments using tableau and MODOC-algorithms.



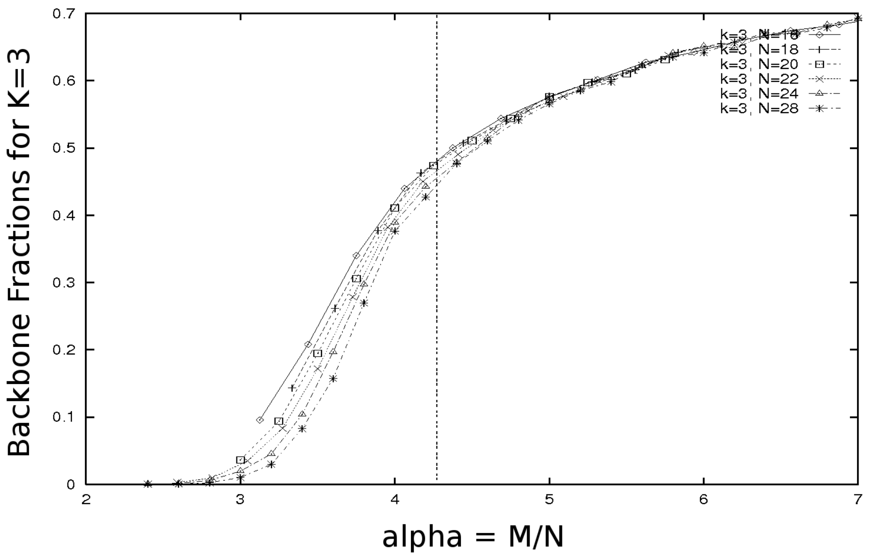


Figure 4: Backbone fractions near phase boundary for K=2 (upper) and K=3 (lower) from [4]

References

[1] D. Mitchell et al., *Hard and Easy Distribution of SAT Problems*, AAAI-92, San Jose, CA, (1992)

[2] J. Crawford et al.,*Experimental results on the crossover point in random 3-SAT*, Artificial Intelligence 81, 31 (1996)

[3] R.Monasson et al. *Statistical Mechanics of the Random K-SAT Model*, arXiv:cond-mat/9606215v2

[4] R. Monasson et al., *2+p-SAT: Relation of typical-case complexity to the nature of the phase transition*, Random Structures and Algorithms 15, 414 (1999)

[5] Von Gelder, A. *Autarky Pruning in Propositional Model Elimination Reduces Failure Redundancy*, Journal of Automated Reasoning 23, 137(1999)