Phase Transition Behavior: from Decision to Optimization

John Slaney
Australian National University
jks@arp.anu.edu.au

Toby Walsh
University of York
tw@cs.york.ac.uk

February 6, 2002

1 Introduction

Combinatorial optimization problems (finding solutions that minimise a cost parameter) are closely related to the corresponding decision problems (deciding whether solutions within a given cost exist). Indeed, many algorithms for optimization essentially work by solving a sequence of decision problems. We might hope, therefore, that insights into decision problems gained by studying phase transition behavior could be useful in understanding similar behavior in optimization problems [ST98, SW01, Zha01]. For both theoretical and historical reasons, propositional satisfiability (or SAT) is the most intensively studied such decision problem. Accordingly, we begin our investigation with the simplest optimization versions of SAT: in the overconstrained case, MAXSAT and in the underconstrained case MAXONES. Given the insights gained from studying the 2+p-SAT decision problem [MZK$^+$99], this paper looks at optimization versions of random k+p-SAT.

2 2+p-SAT

As is well known, there is a sharp transition in satisfiability for random 2-SAT at $l/n = 1$ [CR92, Goe92], and for random 3-SAT around $l/n \approx 4.3$ [MSL92]. Associated with this transition is a rapid increase in problem difficulty. The random 2-SAT transition is continuous (or "2nd order") as the backbone (the fraction of variables taking fixed values) increases smoothly in size. On the other hand, the random 3-SAT transition is discontinuous (or "1st order") as the backbone jumps in size at the phase boundary.

To study this in more detail, Monasson et al. introduced the 2+p-SAT problem class [MZK$^+$99]. This interpolates smoothly from the polynomial 2-SAT problem to the NP-complete 3-SAT problem. A random 2+p-SAT problem in

n variables has l clauses, a fraction $(1-p)$ of which are 2-SAT clauses, and a fraction p of which are 3-SAT clauses. This gives pure 2-SAT problems for $p=0$, and pure 3-SAT problems for $p=1$. For any fixed $p>0$, the 2+p-SAT problem class is NP-complete since the embedded 3-SAT subproblem can be made sufficiently large to encode other NP-complete problems within it.

By considering the satisfiability of the embedded 2-SAT subproblem and by assuming that the random 3-SAT transition is at $l/n \approx 4.3$, we can bound the location of the random 2+p-SAT transition to:

$$1 \leq \frac{l}{n} \leq \min(\frac{1}{1-p}, 4.3)$$

Surprisingly, the upper bound is tight for $p \leq 2/5$ [AKKK01]. That is, the 2-SAT subproblem *alone* determines satisfiability up to $p = 2/5$. Asymptotically, the 3-SAT clauses do not determine if problems are satisfiable, even though they determine the worst-case complexity. Several other phenomena occur at $p = 2/5$ reflecting this change from a 2-SAT like transition to a 3-SAT like transition. For example, the average cost to solve problems appears to increase from polynomial to exponential both for complete and local search algorithms [MZK+98, SGS00]. As a second example, the transition shifts from continuous to discontinuous as the backbone jumps in size [MZK+98]. Random 2+p-SAT problems thus look like polynomial 2-SAT problems up to $p = 2/5$ and NP-complete 3-SAT problems for $p > 2/5$.

3 MAX 2+p-SAT

We begin by looking at the MAX 2+p-SAT optimization problem. We use a two-phase exact algorithm for MAXSAT which runs the GSAT heuristic to generate an initial solution, and then uses a Davis-Putnam style branch and bound algorithm [BF99]. As it is NP-hard to approximate the answer to both MAX 2-SAT and MAX 3-SAT to within any ϵ, it will also be NP-hard to approximate the answer to MAX 2+p-SAT.

In Figure 1, we see that the number of unsatisfied clauses drops linearly with p, and the gradient steepens as l/n increases. We can plot contours in the p against l/n space of problems with an equal number of unsatisfied clauses. These contours are very similar in shape to the cuve separating satisfiable 2+p-SAT decsion problems from unsatisfiable ones. Search cost for MAX 2+p-SAT increases as l/n increases or as p decreases. The hardest MAX 2+p-SAT problems at fixed l/n are for $p = 0$ (i.e. MAX 2-SAT problems). This is perhaps a little surprising. We might expect MAX 3-SAT problems to be more difficult than MAX 2-SAT problems, as MAX 3-SAT problems contain 3-clauses, which are typically more difficult to reason with than 2-clauses (e.g. they give less unit propagation). However, the number of unsatisfied clauses is largest for $p = 0$ and this dominates the search cost.

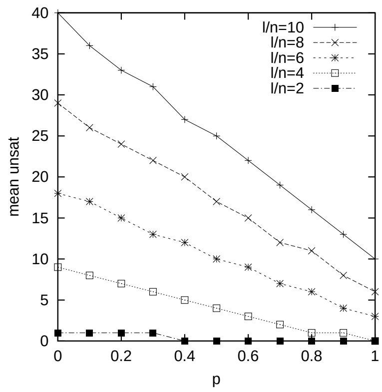

Figure 1: Number of unsatisfied clauses in a MAX 2+p-SAT problem. 100 problems are generated at each point, $p = 0$ to 1 in steps of 0.1 and $n = 30$.

4 MAX 1+p-SAT

The 2+p-SAT problem interpolates smoothly from a polynomial decision problem to an NP-complete decision problem. To interpolate smoothly from a polynomial optimization problem to one that is NP-hard to approximate, we introduce the 1+p-SAT problem class. A random 1+p-SAT problem in n variables has l clauses, a fraction $(1 - p)$ of which are 1-SAT clauses, and a fraction p of which are 2-SAT clauses. The MAX 1+p-SAT problem is to find the maximum number of satisfied clauses in a 1+p-SAT problem. MAX 1-SAT is polynomial to solve, just by counting pairs of complementary literals. On the other hand, for any fixed $p > 0$, MAX 1+p-SAT is NP-hard to approximate. The number of unsatisfied clauses for MAX 1+p-SAT behaves in a similar way to MAX 2+p-SAT. However, Figure 2 highlights a critical difference in computational complexity. At fixed l/n, the hardest MAX 1+p-SAT problems are at $p \approx 0.7$. Surprisingly MAX 1.7-SAT appears harder than MAX 2-SAT. That is, adding a polynomial subproblem makes an NP-hard optimization problem orders of magnitude harder. How can this be? In fact, it illustrates the difference in "flavor" between optimization and decision: whilst decreasing p moves us closer to a purely polynomial problem, it also increases the number of unsatisfied clauses. There is a tradeoff between constrainedness (unsatisfiability) and simplicity. The hardest problems are therefore at an intermediate value of p.

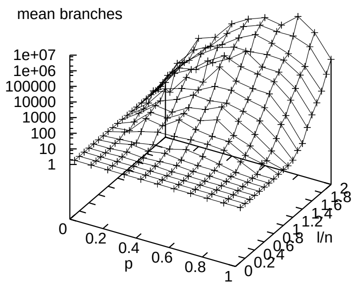

Figure 2: Mean branches to solve a MAX 1+p-SAT problem (y-axis logscale) against p (x-axis). 1000 problems are generated at each point, $p = 0$ to 1 in steps of 0.1, and $n = 100$.

5 MAXONES

Another optimization problem derived from the SAT decision problem is MAXONES. The aim is to find a satisfying truth assignment that that maximizes the number of variables assigned the value 1 (or true). This is a special case of DISTANCESAT: find a satisfying truth assignment that diagrees as little as possible with a given partial interpretation of the variables [BM99]. DISTANCESAT has important practical applications: for instance, many CSPs with soft constraints can be encoded naturally into DISTANCESAT. Even MAXSAT can be represented as a DISTANCESAT problem: simply replace each clause C_i with $\neg p_i \vee C_i$ where p_i is a variable new to the problem, and try to maximise the number of these new variables set to 1. One reason for studying MAXONES as well as MAXSAT is that it is displays interesting behavior in the underconstrained SAT region, whereas MAXSAT is perhaps most interesting in the overconstrained SAT region.

Figure 3 shows that 2+p-MAXONES follows the same pattern as MAX 2+p-SAT as regards solution size. For a fixed l/v, the greater the proportion of 2-clauses the more constrained the SAT problem, and so the more one is forced to make variables false in order to satisfy it. For fixed p, similarly, the constrainedness increases with the number of clauses.

Figure 4 is more interesting. It shows the computational cost (number of branches explored) of solving 2+p-MAXONES using a simple Davis-Putnam algorithm with branch and bound and with the MOMS heuristic for variable selection. It is clear from the contour plot that the hard problems lie in a diagonal band, so any slice across the plot, either vertical or horizontal, will show a

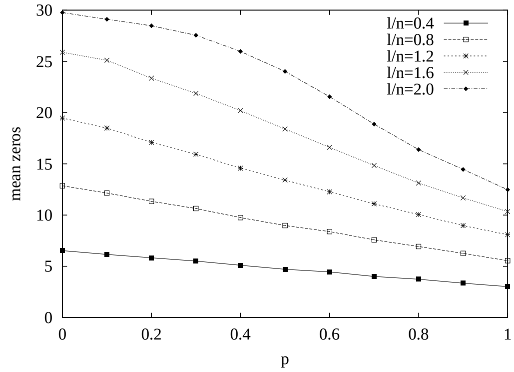

Figure 3: Number of variables assigned 0 in the optimal solution to a 2+p-MaxOnes problem. 1000 problems are generated at each point, $p = 0$ to 1 in steps of 0.1 and $n = 75$.

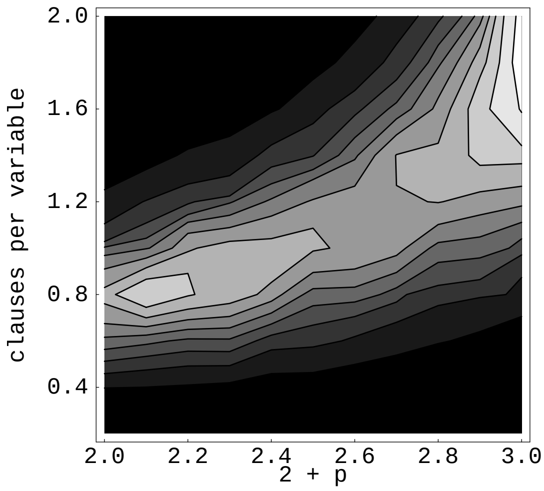

Figure 4: Contour plot of median branches to solve a 2+p-MaxOnes problem against $2 + p$ (x-axis) and l/v (y-axis). 1000 problems are generated at each point, $p = 0$ to 1 in steps of 0.1, and $l/v = 0.2$ to 2 in steps of 0.2. Each point is the median over 1000 problems, $n = 75$. Darker regions represent regions requiring fewer median branches to solve.

cost peak somewhere in the range. Thus, for instance, the cost of solving 2+p-MaxOnes with 2 clauses per variable increases monotonically with p, while for problems with only 0.8 clauses per variable the cost peaks close to $p = 0.1$. 2+p-MaxOnes is therefore like Max 1+p-Sat, and unlike the decision versions k+p-Sat, in that the hardest problems do not always occur where $p = 1$. It is unlike Max 1+p-Sat, however, in that the cost peak in terms of p is not independent of l/v.

Another feature clearly visible in Figure 4 is a saddle point near $p = 0.65$ and $l/v = 1$. It seems that the hardness of 2-MaxOnes-like problems peaks at $l/v \approx 0.8$ and that of 3-MaxOnes-like problems peaks at $l/v \approx 1.8$, while there is a range of 2+p-MaxOnes problems with p between 0.4 and 0.8 which fall into neither class and which exhibit a smaller cost peak. It is unclear exactly why this should be. Nor is it clear why the saddle should be at $p \approx 0.65$ rather than being closer to the transition point $p \approx 0.4$ in the 2+p-Sat decision problem.

As in the case of MaxSat, k-MaxOnes is of polynomial complexity for $k = 1$ and NP-hard for $k = 2$, so again we examined the transition between these cases through 1+p-MaxOnes. This time, however, there were no surprises. The solution size for 1+p-MaxOnes is monotonic in the constrainedness, like that for 2+p-MaxOnes, while the computational cost increases exponentially with p, at least for l/v up to 0.8 which as already noted is the cost peak for 2-MaxOnes.

6 Conclusions

The study of optimization problems related to the 2+p-Sat decision problem throws fresh light on phase transition behaviour and the differences between decision and optimization. There are relationships between average clause length, constrainedness and the hardness of the optimization problems, but these have their own structure which does not seem to mirror the more familiar ones found in the decision problems.

In Max 2+p-Sat, the hardest problems at a fixed l/n are at $p = 0$. In some sense, Max 2-Sat is therefore harder than Max 3-Sat. In Max 1+p-Sat, on the other hand, the hardest problems at a fixed l/n are at $p \approx 0.7$. Max 1.7-Sat is thus typically harder than Max 2-Sat. A polynomial subproblem, by worsening the bound, makes it harder to solve the Max 1+p-Sat problem.

The situation in k+p-MaxOnes is a little different. In 1+p-MaxOnes the hardest problems are at $p = 1$, as might be expected, with no special behavior at $p = 0.7$ or anywhere else. 2+p-MaxOnes, however, shows a more complex pattern: a double cost peak with a saddle between the two peaks. The hardest problems for a fixed p occur at some l/v ratio, but at different ratios for different values of p. Conversely, for a fixed l/v the hardest problems occur at some value of p, but at different values for different ratios. As in the case of Max 1+p-Sat, there is a region (below and to the right of the "ridge" shown in the contour plot) where the more we add of the polynomial subproblem the harder the NP-hard problem becomes on average.

In future research we shall seek ways to transfer other phenomena associated with phase transitions in decision problems such as SAT to their optimization correlates such as MAXSAT and MAXONES. For example, we can gain insight into the performance of the Davis Putnam algorithm on 3-SAT by following trajectories in the 2+p-SAT phase space [CM01]. At each branch point in its backtracking search tree, the Davis Putnam algorithm has a mixture of 2-SAT and 3-SAT clauses. Each branch is thus a trajectory in the 2+p-SAT phase space. Similarly we may be able to understand better optimization algorithms for problems like MAX 2-SAT or 3-MAXONES by tracing trajectories in the MAX 1+p-SAT or 2+p-MAXONES phase spaces.

References

[AKKK01] D. Achlioptas, L.M. Kirousis, E. Kranakis, and D. Krizanc. Rigorous results for (2+p)-SAT. *Theoretical Computer Science*, 265(1-2):109–129, 2001.

[BF99] B. Borchers and J. Furman. A two-phase exact algorithm for MAX-SAT and weighted MAX-SAT problems. *Journal of Combinatorial Optimization*, 2:299–306, 1999.

[BM99] Olivier Bailleux and Pierre Marquis. DISTANCE-SAT: Complexity and algorithms. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99)*, pages 642–647, Orlando, Florida, 1999.

[CM01] S. Cocco and R. Monasson. Trajectories in phase diagrams, growth processes and computational complexity: how search algorithms solve the 3-satisfiability problem. *Physical Review Letters*, 86(8):1654–1657, 2001.

[CR92] V. Chvatal and B. Reed. Mick gets some (the odds are on his side). In *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science*, pages 620–627. IEEE, 1992.

[Goe92] A. Goerdt. A theshold for unsatisfiability. In I. Havel and V. Koubek, editors, *Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science, pages 264–274. Springer Verlag, 1992.

[MSL92] D. Mitchell, B. Selman, and H. Levesque. Hard and Easy Distributions of SAT Problems. In *Proceedings of the 10th National Conference on AI*, pages 459–465. American Association for Artificial Intelligence, 1992.

[MZK+98] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky. Determining computational complexity for characteristic 'phase transitions'. *Nature*, 400:133–137, 1998.

[MZK+99] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky. 2+p SAT: Relation of typical-case complexity to the nature of the phase transition. *Random Structures and Algorithms*, 15(3-4):414–435, 1999.

[SGS00] J. Singer, I.P. Gent, and A. Smaill. Local search on random 2+p-SAT. In *Proceedings of the 14th ECAI*. European Conference on Artificial Intelligence, 2000.

[ST98] J. Slaney and S. Thiébaux. On the hardness of decision and optimisation problems. In *Proceedings of the 13th ECAI*, pages 244–248. European Conference on Artificial Intelligence, 1998.

[SW01] J. Slaney and T. Walsh. Backbones in optimization and approximation. In *Proceedings of 17th IJCAI*. International Joint Conference on Artificial Intelligence, 2001.

[Zha01] W. Zhang. Phase transitions and backbones of 3-SAT and Maximum 3-SAT. In T. Walsh, editor, *Proceedings of 7th International Conference on Principles and Practice of Constraint Programming (CP2001)*. Springer, 2001.