Computational Complexity and Phase Transitions

Peter Cheeseman
RIACS*

Bob Kanefsky
Sterling Software

William M. Taylor
Sterling Software

Artificial Intelligence Research Branch
NASA Ames Research Center, Mail Stop 269-2
Moffett Field, CA 94035, USA
Email: <last-name>@ptolemy.arc.nasa.gov

Abstract

It is well known that for many NP-complete problems, such as K-Sat, K-colorability etc., typical cases are easy to solve; so that computationally hard cases must be rare (assuming P $\neq$ NP). This paper shows that NP-complete problems can be summarized by at least one "order parameter", and that the hard problems occur at a critical value of such a parameter. This critical value separates two regions of characteristically different properties. For example, for K-colorability, the critical value separates overconstrained from underconstrained random graphs, and it marks the value at which the probability of a solution changes abruptly from near 0 to near 1. It is the high density of well-separated almost solutions (local minima) at this boundary that cause search algorithms to "thrash". This boundary is a type of phase transition and we show that it is preserved under mappings between problems. We show that for some P problems either there is no phase transition or it occurs for bounded N (and so bounds the cost). These results suggest a way of deciding if a problem is in P or NP and why they are different.

1 Introduction

A common result of AI research is to show that some class of problems is NP-complete (or NP-hard), with the implication that this class of problems is very hard to solve (assuming P $\neq$ NP). On the other hand it is well known that for many of these NP problems, typical instances are easy to solve (e.g. [12]). There is no contradiction here, since NP complexity is usually a worst case analysis for a whole class of problems, and so says nothing about the difficulty of typical instances. However, this situation raises the question "where are the really hard instances of NP problems?". Can a subclass of problems be defined that is typically (exponentially) hard to solve, or do worst cases appear as rare "pathological cases" scattered unpredictably in the problem space?

In this paper we show that for many NP problems one or more "order parameters" can be defined, and hard instances occur around particular critical values of these order parameters. In addition, such critical values form a boundary that separates the space of problems into two regions. One region is underconstrained, so the density of solutions is high, thus making it relatively easy to find a solution. The other region is overconstrained and very unlikely to contain a solution. If there are solutions in this overconstrained region, then they constitute such deep minimuma (strong basins of attraction) that any reasonable algorithm is likely to find it. If there is no solution, then a backtrack search can usually establish this with ease, since potential solution paths are typically cut off early in the search. Really hard problems occur on the boundary between these two regions, where the probability of a solution is low but non-negligible. At this point there are typically many local minima corresponding to almost solutions separated by high "energy barriers". These almost solutions form deep local minima that usually trap search methods that rely on local information.

Because it is possible to locate a region where hard problems occur, it is possible to predict whether a particular problem is likely to be easy to solve. We expect that in future computer scientists will produce "phase diagrams" for particular problem domains to aid in hard problem identification and for prediction of solution existence probability, such as shown in [8].

We present these ideas by first showing how phase transitions arise in problem solving, and then illustrating particular transitions through several examples with different properties. We then show how some of these examples interrelate when they are mapped onto each other. Finally, we summarize the results and state a strong conjecture based on these results.

2 Phase Transitions

We first review well-studied cases where the behavior of a complex system, including phase transitions, can be described by an order parameter. For example, the probability that a random graph is connected, or contains a Hamilton circuit, or a triangle etc., has a sharp threshold for particular values of the average graph connectivity [1]. In the case of graph connectivity and Hamilton circuits, this threshold depends on the graph size. Other

*Research Institute for Advanced Computer Science

0-8186-3420-0/93 $03.00 © 1993 IEEE

properties of random graphs also show interesting behavior around the transition point which is characteristic of phase transitions. In particular, the size of the largest connected sub-graph grows very rapidly as a function of the average connectivity as the critical connectivity is approached from below. Also, the sizes of the subgraphs below the threshold show a fractal distribution—these properties are related to analogous physical systems, e.g. [9].

Our interest in phase transitions arises from the discovery that hard to solve problems occur at such boundaries for many types of problems. The importance of phase transitions for AI is discussed in [5] where it is argued that complex systems composed of many interacting values can often be understood at the macroscopic level in terms of a few order parameters that are characteristic of the system as a whole. Summarizing the properties of complex systems through a small set of parameters is routine in statistical mechanics [2],[9]. This is possible because a large number of local interactions can produce dramatic coordinated macroscopic behavior, such as phase transitions, that do not depend on the detailed interactions within the system. Examples of phase transitions in AI are given in [7],[8].

3 An Example: Graph Coloring

This is a constraint satisfaction problem, where each variable can take on a number of possible values ("colors"), and there are binary constraints that forbid particular pairs of variables from having the same color. The goal is to see if there is an assignment of colors to the variables that satisfy the constraints and only use K colors, or report that no assignment is possible. An alternative goal is to find the minimum K that satisfies the constraints (the chromatic number problem). Any solution to a graph coloring problem can be used to generate other solutions by interchanging the colors, implying a color rotation symetry. Many practical constraint satisfaction problems, such as timetable construction, can be mapped into a graph coloring problem.

3.1 Empirical Results

Graph coloring has been extensively investigated, both theoretically and empirically, e.g., [11],[12]. Even though graph coloring is an NP-complete problem, these authors report that graph coloring is "almost always easy". In particular, a simple backtrack algorithm by Brelaz was found to solve all randomly generated graphs it was tried on with little backtrack [12]. We continue these investigations but restrict our attention to random graphs that have been "reduced". The "reduction operators" decribed below guarantee that if the reduced graph is K-colorable (or not) then the original graph is K-colorable (or not). Any graph that can be reduced to one that is trivially K-colorable (or not) can be solved without search. We only investigate the space of reduced graphs, because the hard problems must be in this space. The particular reduction operators we used for K-colorability are:

1. Remove Underconstrained Nodes—a node with less than K constraints can be removed, because it can always be colored, regardless of the colors of its neighbors.
2. Remove Subsumed Nodes—a node N can be removed if there is a node M that is connected to everything N is connected to, since any color that works for M will work for N (provided N is not connected to M).
3. Merge nodes that must have the same color—if any nodes are fully connected to a clique of size K-1 (but not to each other), then these nodes must have the same color, and so can be merged.

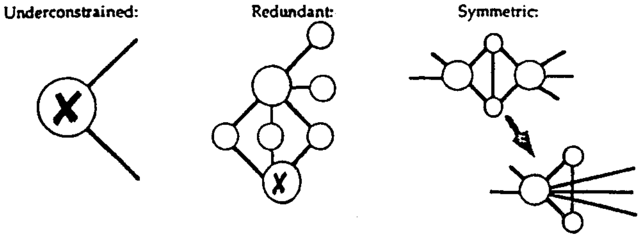

Figure 2: Reduction Operators for 3-Colorability

These reduction operators can be applied in any order, and typically the application of one operator creates a situation where other operators become applicable, producing a reduction cascade. We found that these operators reduced all our carefully hand-constructed "hard" graphs to trivial cases! Although these are all the single node reduction operators we could find, multi-node reduction operators do exist. The essential difference between problem reduction and problem solving is that problem reduction does not produce disjunctive alternatives (i.e. no search). The following investigations are all in the space of random reduced graphs, because the K-colorability of a reduced graph is equivalent to that of many unreduced graphs.

We empirically investigated the probability of a solution for K-colorability problems for different values of K and N (number of nodes). The results are shown in Fig. 2a, where each probability point is the average of about 5 trials, but there are no points in the transition region because they are too costly to compute. Two trends are clear from these results. First is the abrupt change in solution probability occurs at higher values of the connectivity for larger K, and the other is the sharpness of the transition increases with N.

We next show how the computational cost of running the Brelaz algorithm varies as a function of the connectivity for different values of K. The results are shown in Figs. 2b,c for random reduced graphs that were generated so that they were guaranteed to have a solution. For both 3-Col and 4-Col the existence of a phase transition is clear, and their location is the same as that for the corresponding solution-probability transition to within the numeric noise level. The transition for 4-Col is much sharper than for 3-Col. Similar results are obtained for random reduced graphs that are not guaranteed to have a K-col solution, and for random graphs restricted to 2-D neighbor connections.

Brelaz's algorithm [12] uses heuristics: Select an uncolored node with the fewest remaining colors; ties are broken by selecting the node connected to the most uncolored nodes; remaining choices are made randomly. This is a very effective algorithm, but its performance at the phase boundary is highly variable, even if rerun on the same graph, because the randomness in the heuristics. These "fluctuations" are typical of behavior near a phase transition. This observation suggests an improvement for a backtrack algorithm—run many versions of it in parallel, so that the expected number of steps is lower than for a single version. These results show that there is a phase transition for the cost of solving K-colorability problems, and it occurs at the critical average connectivity[1] where the probability of a solution has dropped to almost zero. This explains why previous authors [11],[12] found K-colorability an easy problem—they were using *nonreduced* graphs whose effective connectivities did not typically fall near the phase boundary.

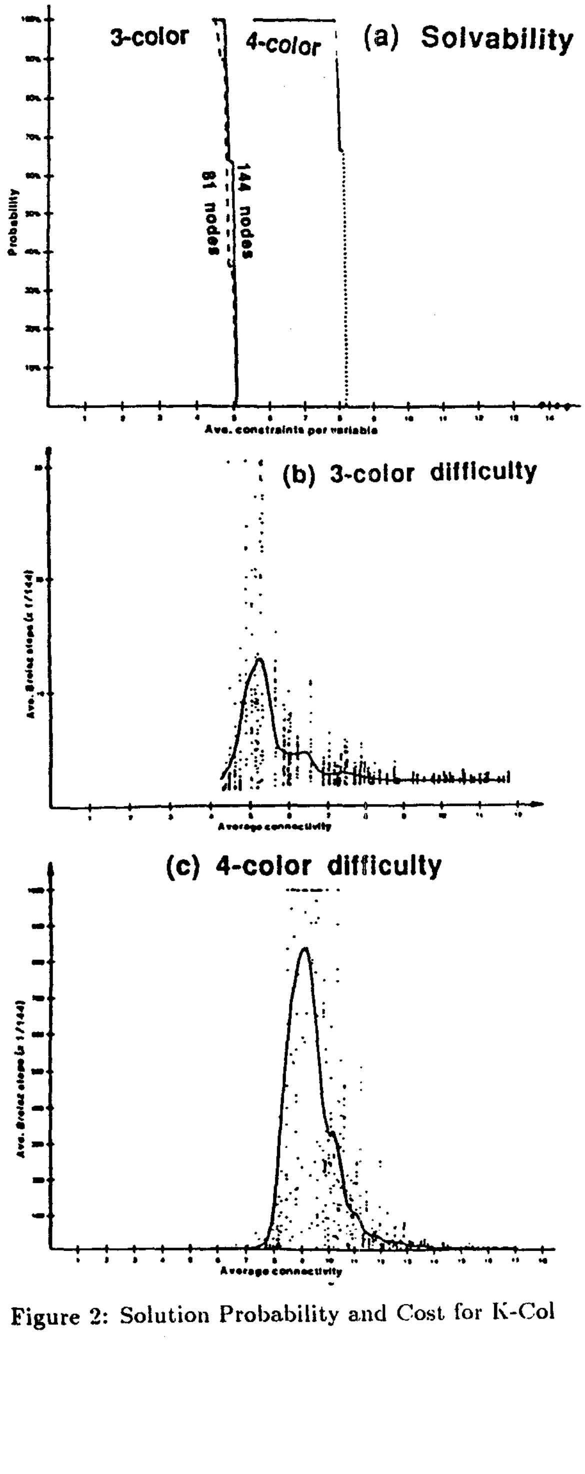

Figure 2: Solution Probability and Cost for K-Col

Examination of Brelaz's algorithm on the hard instances shows that it often backtracks, and sometimes backtracks all the way to the beginning. This "thrashing" occurs because there are many near-solutions available, and they look like potential solutions to the backtrack algorithm until an assignment is nearly complete. These local minima make it hard to find a solution if one is present—the proverbial needle in a haystack. This observation hints as to why some NP instances are hard; it seems likely that any algorithm based on local information will be fooled by the high number of local minima in problems near the phase boundary. Further experimentation suggests that graphs with a high variance of the average connectivity are generally easier to solve than ones with lower variance, so that the variance of the average connectivity may be an additional order parameter.

An apparent exception to the conjecture that all the NP problems overlap the critical boundary, is to discover if planar graphs with node connectivity ≤ 4) are 3-Col. This is an NP-complete problem [4], yet our 3-Col transition for random reduced graphs occurs at a connectivity ≈ 5.4 (see Fig. 2b)). To compare planar graphs with our results we must first reduce them. Merging destroys the planarity of the graph, reduces the number of nodes and simultaneously increases the connectivity of the merged nodes. The results is increased average connectivity of the previously planar graphs so that they straddle the critical connectivity, thus preserving the conjecture. However, if the proportion of 4-connected nodes in 3,4 planar graphs are sufficiently restricted, the result is a new P class of problems, assuming the conjecture is true.

3.2 Analytic Results—K-Col

The location of the phase boundary for graph coloring can be found approximately by the following simple argument:

Let N be the number of nodes a a graph; E be the number of edges in the graph, and K be the number of colors available. For a random coloring, the probability that a particular edge does not violate the color equality constraint is: $(K-1)/K$ [i.e. the probability that the nodes at either end of the edge are not the same color]. If conditional independence of the edges succeeding is assumed, the probability that *all* edges succeed is:

$$\text{Probability a coloring is solution} = (\frac{K-1}{K})^E. \quad (1)$$

[1] Average connectivity or vertex degree is defined here to be the average number of constraints (edges) incident on a node. Because a binary constraint involves 2 nodes; average connectivty = 2 x (number constraints)/(number nodes)—this is a different definition than used in [6].

Since there are K^N possible colorings, the total expected number of solutions is:

$$\text{Expected number of solutions} = (\frac{K-1}{K})^E K^N. \quad (2)$$

Note that the first term rapidly *decreases* as E increases, while the second term rapidly *increases* with increasing N. This competion between opposing terms is responsible for the phase transition. Continuing the analysis, we have the probability that a random color assignment of an edge will succeed is given by: $\frac{K-1}{K}$, and so the probability *all* edges in a random assignment will succeed is given by: $(\frac{K-1}{K})^E$, so assuming conditional independence of assignments, we have:

$$\text{Probability no Soln} = P_0 = [1 - (\frac{K-1}{K})^E]^{K^N} \quad (3)$$

Some algebra shows that:

$$P_0 \approx \exp(-(\frac{K-1}{K})^E K^N) = \exp(- < \# \text{ solns} >). \quad (4)$$

A graph of this function looks very similar to Fig. 2— i.e. there is a sharp transition in solution proability at a critical value of average connectivity.

An analytic form of the critical connectivity can be derived by assuming the turning point occurs when $P_0 = e^{-1}$. Under this assumption, the critical average connectivity is given by:

$$\text{Critical Connectivity} = C_0 = \frac{2 \ln K}{\ln \frac{K-1}{K}}. \quad (5)$$

For $K = 3$, $C_0 = 5.42$, and for $K = 4$, $C_0 = 9.64$. Both these values agree with the empirical results in Figs. 2 and 3 to within the experimental error.

The approximate value for C_0 depends strongly on the assumption that the assignments are (conditionally) independent and that the probability of an edge not being in violation is independent of the knowledge that the other edges are not in violation. Both these assumptions are false in general, but in most circumstances they are a good approximation. For example, the knowledge that a is connected to b and b to c and that these edges are not in violation induces a correlation between the colors on a and c, so that the probability that they are not the same color is given by $(K-2)/(K-1)$ rather than $(K-1)/K$. However, for longer circuits, the probability of the last closing edge being in violation is very nearly the independent value. These observations explain why planar graphs are an exception to the approximate critical value above: in planar graphs there are many circuits of length 3 so the edge independence assumption is not even approximately correct. For random graphs, however, the average circuit size is large, so that the above critical value is quite accurate.

This approximate approach can be extended to give the slope of P_0 at the critical value. The result is that the slope is proportional to N and $Log(\frac{K}{K-1})$. In other words the phase transition gets sharper with larger N, as expected; but the dependence on K is weak.

The analytic results reported above are more accurate than a different approximate analysis [13] which made different independence assumptions. It may be possible to develop better approximations by taking conditional dependencies into account by a perturbation analysis on the formulas derived above.

4 An Example: K-Satisfyability

Since it is possible to map K-colorability problems into K-sat problems and vice versa [4], this implies that K-Sat and K-Col are essentially the same problem. Because of this mapping, we found, as expected, similar phase transitions in K-Sat.

4.1 Empirical Results

We used a form of resolution to reduce random K-sat problems before applying a simple backtrack search procedure with a most-constrained-first heuristic. This backtrack search procedure maximally reduced the current problem before chosing another variable to assign. The most important reduction consisted of immediately substituting any variable whose value was forced, and propagating the consequences. This algorithm is similar to the Putnam-Davis algorithm [6]. The results are shown in Fig. 3. As for K-colorability, there is a sharp drop in the probability of a solution at some critical value of the graph average connectivity, and this critical value depends on K. Also, the normalized cost of solution shows a phase transition at about the point at which the solution probability drops to near zero. Similar results were found for random reduced graphs whose method of construction guaranteed at least one solution. For 2-Sat we also found a "bump" in the computational cost around values of the average connectivity at which the solution probability drops rapidly to zero. However, this bump does not grow exponetially with N (the number of variables), as expected, since 2-Sat is a polynomial time problem.

4.2 Analytic Results—K-Sat

An analysis similar to that for K-Col can be given for K-Sat. Let N be the number of (binary) variables; M be the number of clauses (constraints); and L be the number of variables (negated or not) in each clause. Using these definitions and (conditional) independence assumptions, we get the following for the probability of no solution:

$$P_0 \approx e^{-(\frac{2^L-1}{2^L})^M 2^N} = e^{-<\# \text{ solns}>}. \quad (6)$$

From this equation we derive the following critical value for the average node connectivity:

$$\text{Critical Connectivity} = C_0 = \frac{L \ln 2}{\ln \frac{2^L-1}{2^L}}. \quad (7)$$

The numeric value for C_0 for $L = 3$ is 15.6, compared to an experimental value of about 12-16. A recent empirical study based on more data gave a result of about 12.9 [6]. This 12.9 result is sufficiently different[2] from the approximate analytic value that the difference must be due to a partial breakdown of the independence assumptions. For $L = 4$, C_0 is 32.3, which is in the range 30-40 observed empirically in Fig 3.

[2] Reference [6] used average connectivity M/N instead of average node connectivity $L * M/N$.

5 Other Examples

We have investigated other problems that differed significantly from either K-Sat or K-Col[2]. In particular, we investigated Hamiltonian circuit and found a sharp phase transition at just the value of average connectivity where the probability of existance of a Hamilton circuit rises from zero to one [3]. This result is interesting because Hamilton circuit problems have a single connectedness constraint involving *all* the variables instead of a few variables as in the previous examples.

Another problem we investigated[2] was the travelling salesman problem with the order parameter being the variance of the cost matrix (integer costs). As with the other examples, we discovered a phase transition in problem complexity for a particular critical value of the variance.

6 Mappings Between Problems

Perhaps the main contribution of the NP-completeness theory is the demonstration that many apparently different problems can be mapped into each other so that solutions are preserved under the mapping. The main conjecture of this paper is that problems whose order parameter is at the critical boundary are typically really hard. If this is true, then an important question is whether the critical boundaries are preserved under these problem mappings, as would be expected if this conjecture is true. We found this to be the case for K-Sat and K-Colorability problems.

Another interesting question is what happens to a P problem if it is mapped into an NP problem in the same family? Do P problems avoid the critical region in such a mapping? As an example, consider the mapping of 2-sat (a P problem) into 3-sat (an NP problem). Such a mapping is given in [4], where a 2-sat clause, such as $(a \vee \bar{b})$ goes into two clauses $(a \vee \bar{b} \vee x)$ and $(a \vee \bar{b} \vee \bar{x})$. Since every such transformation introduces an extra variable (e.g. x) which only occurs in two clauses, the average connectivity of all the variables is dragged below the critical threshold. In other words, just transforming 2-sat into 3-sat by trivial variable addition does not produce hard problems, since the transformed problems do not overlap the critical region.

In view of these results it is tempting to conjecture that the difference between P and an NP problems is whether a phase boundary exists or not. Unfortunately, this is not true—what matters is whether the phase boundary (if there is one) occurs at a fixed N or not. To explain this distinction, we compare the above results with the N-Queens problem [11], which is a known to be P. For N-Queens, there is a phase transition at $N \approx 10$, but because this phase transition occurs for a fixed (low) N, the amount of computation is strongly bounded, as expected for a P problem.

7 Discussion

Because of the basic equivalence of NP-complete problems, we expect phase boundaries in NP complete problems other than those investigated here. However, constrained minimization problems such as graph partition, integer partition, maximal clique, Ramsey numbers etc., may have a different order parameters. Some of these problems show a "spin-glass like" transition [2].

An objection to the above results may be that they are all based on heuristic backtrack search, so that the apparent phase transition in computational cost may be a result of this choice rather than intrinsic to the problem. For graph coloring, a "local repair" algorithm [11] and a probablistic search procedure also had difficulties with reduced graphs in the same critical connectivity range, adding confirmation that the phase phenomenon is intrinsic. The difficulties experienced by all these algorithms seems to be due to the large number of local minima and the high "energy barriers" between them—a property of the problem and not dependent on a particular algorithm.

8 Conclusions and Conjectures

The results reported above suggest the following conjecture:

> All NP-complete problems have at least one order parameter and the hard to solve problems are around a critical value of this order parameter. This critical value (a phase transition) separates one region from another, such as overconstrained and underconstrained regions of the problem space. In such cases, the phase transition occurs at the point where the solution probability changes abruptly from almost zero to almost 1.

The converse conjecture is:

> P problems do not contain a phase transition or if they do it occurs for bounded N (and so has bounded cost).

We have presented empirical evidence for these conjectures and approximate analytical values for particular problem classes, and have shown in some cases that the hard problems in one space map into hard problems in the other space, thus preserving the phase boundary under the mapping. If these conjectures are true, then all that is needed to turn an NP problem into a P problem is to add restrictions so problems near the critical value of the order parameter are excluded.

There are many outstanding questions, such as: "What happens for NP-hard problems"; "Can hard problems occur in the non-critical region?", "Do other types of problems, such as optimization problems, games, etc. have the same properties?"; and so on.

Acknowledgements

We wish to gratefully acknowledge the many stimulating discussions and ideas of W. Buntine, E. Gamble, S. Minton, A. Philips, A. Mayer, O. Hansen and B. Pell.

References

[1] Bollobas, B. "*Random Graphs*", Academic Press, London, 1985.

[2] Cheeseman, P., Kanefsky, B., and Taylor, W., "it Where the Really hard Problems Are", Proc. 12th. International Joint Conference on Artificial Intelligence (IJCAI-91), pp 331-337, 1991.
[3] Fu, Y. "*The Uses and Abuses of Statistical Mechanics in Computational Complexity*", in " Lectures in the Sciences of Complexity", Ed. D. L. Stein, pp 815-826, Addison Wesley, 1989.
[4] Garey M. R. and Johnson D. S., "*Computers and Intractability': A Guide to the Theory of NP-Completeness*", Freeman, 1979.
[5] Huberman, B. A. and Hogg, T., "*Phase Transitions in Artificial Intelligence Systems*", Artificial Intelligence, 33, 155-171, 1987.
[6] Michell, D., Selman, B. and Levesque, H., "*Hard and Easy Distributions of Sat Problems*", Proc. 10th. National Conference on Artificial Intelligence (AAAI-92), pp 459-465, 1992.
[7] Karp, R. M. and Pearl, J., "*Searching for an Optimal Path in a Tree with Random Costs*", Artificial Intelligence,21, (1,2), 99-116, 1983.
[8] Purdom, P. W., "*Search Rearrangement Backtracking and Polynomial Average Time*", Artificial Intelligence,21, (1,2), 117-134, 1983.
[9] Kirkpatrick, S. and Swendsen, R. H., "Statistical Mechanics and Disordered Systems", Comm. ACM, 28, 4, 363-373, April 1985
[10] Little, J. D. C., et al., "*An Algorithm for the Traveling Salesman Problem*", O.R.S.A., 11, 972-989, 1963.
[11] Minton, S. et al, "*Solving Large-Scale Constraint Satisfaction and Scheduling Problems Using a Heuristic Repair Method*", Proc. 8th. Nat. Conf. on A.I. (AAAI-90), 17-24, 1990.
[12] Turner, J. S., "*Almost All k-Colorable Graphs are Easy to Color*", Journal of Algorithms, 9, 63-82, 1988.
[13] Wilson, C. and Hogg, T., "*Using Deep Structure to Locate Hard Problems*", Proc. 10th. National Conference on Artificial Intelligence (AAAI-92), pp 472-477, 1992.

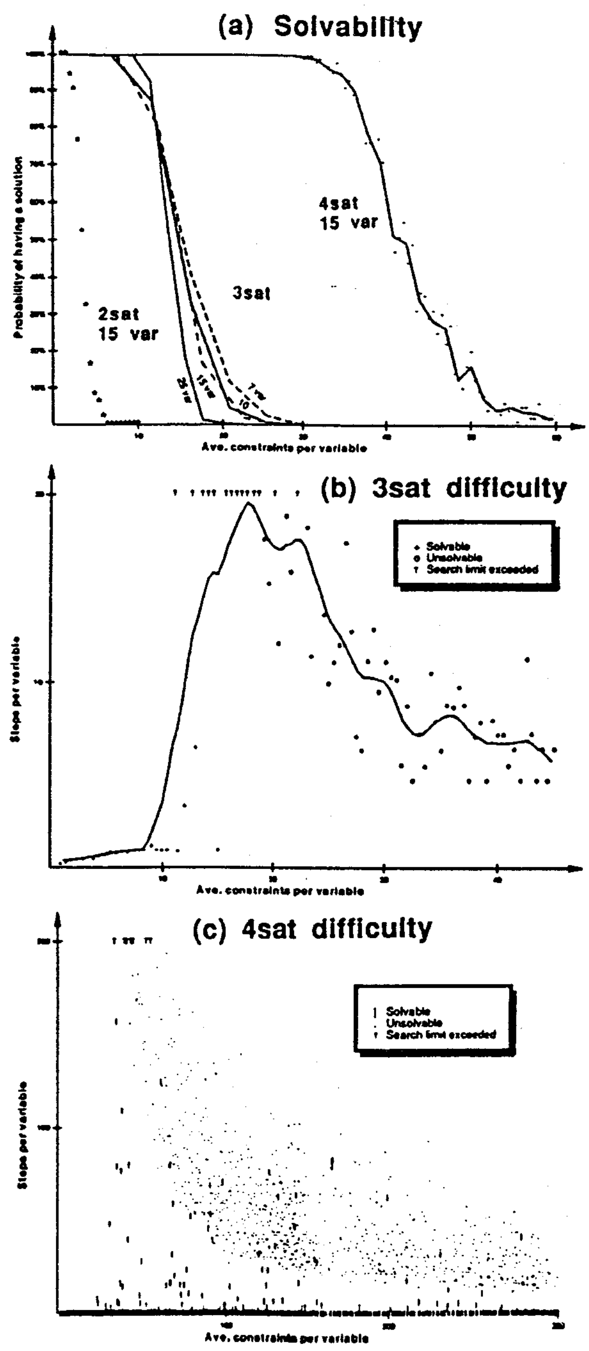

Figure 3: Solution Probability and Cost for K-sat