Exploiting the Deep Structure of Constraint Problems

Colin Williams Tad Hogg

Artificial Intelligence **70**, 73–117 (1994)

Abstract

We introduce a technique for analyzing the behavior of sophisticated A.I. search programs working on realistic, large-scale problems. This approach allows us to predict where, in a space of problem instances, the hardest problems are to be found and where the fluctuations in difficulty are greatest. Our key insight is to shift emphasis from modelling sophisticated algorithms directly to modelling a search space that captures their principal effects. We compare our model's predictions with actual data on real problems obtained independently and show that the agreement is quite good. By systematically relaxing our underlying modelling assumptions we identify their relative contribution to the remaining error and then remedy it. We also discuss further applications of our model and suggest how this type of analysis can be generalized to other kinds of A.I. problems.

Chapter 1

Introduction

In recent years, there has been a wave of experimental results reporting how different constraint satisfaction algorithms typically perform across a range of problem instances [35, 13, 4, 30, 26]. These studies reveal the existence of an easy-hard-easy pattern in computational cost, as a parameter, that distinguishes different classes of problem instances, is varied. The fact that this pattern recurs so often, across different problems, solved using different algorithms, suggests that the explanation for it lies more with the problem than with any algorithm. In this paper we therefore present an analysis of the deep structure of constraint *problems* that explains both the qualitative existence of the observed variation in difficulty and how to make quantitative estimates of quantities of interest such as the location of the hardest problems and the probability of having any solutions at all.

There are two key insights that allow us to do this. The first is to switch from modelling the behavior of the sophisticated algorithm directly, to modelling a representative search space that a sophisticated algorithm might navigate. Such a space can then be viewed as an underlying deep structure of the problem, and can be characterized by interpreting "sophistication" as being some mechanism whereby redundant and irrelevant search is avoided. A particular example is the lattice of assumptions that assumption-based truth maintenance systems (ATMSs) navigate [9]. This allows us to finesse handling the minutiae of real algorithms and yet still make quantitatively accurate predictions concerning real problems.

The second key ingredient is the observation that it is not necessary to specify the deep structure in complete detail before reliable estimates of properties of interest can be made. This is fortunate because realistic systems, e.g., those that interact with complex and unpredictable environments such as the physical world [7], may defy exact specification. In our approach only a few properties of the problem are specified and we assume the others take on values according to some probability distribution, allowing the law of large numbers to give quantitative predictions. Even when a complete specification is available such an approach is valuable in identifying the most important properties of a problem. For example, the lattice of assumptions can be characterized by a particular kind of set system, called a Sperner system, whose description can in turn be "summarized" in just a few parameters. In the interests of an economical model we use as few degrees of freedom, or "order parameters", as possible, keeping just enough to accurately predict the properties of interest. By starting off parsimoniously and only adding more details as they become necessary we discover which aspects of the deep structure are the most important. In any particular case, the adequacy of the choice of the few specified degrees of freedom must be evaluated empirically: major differences with the predictions suggest that additional degrees of freedom must be specified, i.e., for the purposes at hand they are not behaving randomly enough[1].

These considerations have led to a number of studies of large-scale behavior of various problems based on specifications involving a small number of parameters [4, 19, 17, 37, 39] as well as applications of such results [15, 18]. However, these studies have all focussed on the superficial description of a problem and not on its deep structure. The danger in such an oversight is that superficially different problems might in fact be identical because they induce isomorphic deep structures. By using deep structure we can identify truly similar systems and not just superficially similar ones.

We present an analysis of the deep structure of constraint satisfaction problems (CSPs), in terms of

[1] Often some correlations in the unspecified degrees of freedom can be tolerated without major effect on the results [41].

their underlying Sperner systems, and examine its consequences. These include specific phase transitions, in which global properties such as problem difficulty, change abruptly as simple measures of deep structure are varied. Unlike previous predictions of phase transitions in search, our analysis involves easily measurable quantities allowing us to make quantitative comparison with experimental data on real CSPs obtained independently [4, 26]. The results show our simple model predicts the location of the phase transition point quite well.

In contrast to previous studies we also examine the effects of relaxing some of the simplifying assumptions underpinning our theory. These results show that the qualitative phenomena still prevail although the quantitative values of the location of transition points and problem difficulty will change.

We should stress that the analysis we present does not apply to all kinds of algorithms for solving constraint satisfaction problems. One could imagine a spectrum of algorithms of increasing sophistication. At one extreme, there would be hopelessly inefficient methods that took no advantage of memory or problem structure, whilst at the other there would be ultra-sophisticated methods that proceeded directly to a solution without performing any search whatsoever. It is our belief that for realistic, large, problems it is too inefficient to use naive algorithms and too hard to find ultra-sophisticated algorithms that work well universally. In between these extremes, there will be algorithms that have to perform some amount of search, but can do so efficiently. Our analysis will be found to be most applicable to these cases.

The paper is organized as follows: §2 explains how the deep structure of a CSP can be identified with a special lattice structured search space. It also discusses how some well known CSPs scale up as the size of the problems increases. In §3 we explain a proxy search cost measure to roughly characterize the computational effort required to solve the CSP. This section contains the main derivation of the paper. We compare our theory against experimental data in §4. In §5 we carefully test the assumptions underlying our model and in §6 we discuss additional applications. §8 summarizes our discoveries and concludes with an outlook for the future directions in which this work could proceed.

Chapter 2

The Deep Structure of CSPs

2.1 Lattices of Assumptions

For CSPs the deep structure is a fundamental lattice underlying many kinds of problems which compactly encodes the states of the problem in such a way that it becomes possible to avoid redundant and irrelevant search. Within this lattice, certain minimal nodes completely determine the properties of the lattice and hence those of the corresponding problem. These minimal nodes form a mathematical structure known as a *Sperner system*, i.e., a set system such that no set is a subset of any other [3]. In this sense a CSP, at its deepest level, becomes fundamentally equivalent to its Sperner system.

To illustrate how lattice representations arise [11], recall that a constraint satisfaction problem (CSP) involves a set of μ variables, $\{v_1, v_2, \ldots, v_\mu\}$ each having an associated set of domain values, $b_{v_1}, b_{v_2}, \ldots, b_{v_\mu}$ respectively, together with a set of ν constraints, $\{c_1, c_2, \ldots, c_\nu\}$ specifying which assignments of values to variables are compatible ("good") and incompatible ("nogood"). These pairings of a variable with a value are known as "assumptions" and the goal is to find a conjunction of exactly μ assumptions such that every variable has some unique value and all the constraints are simultaneously satisfied. The Sperner system associated with this lattice is the set of minimized nogoods, i.e., the sets of incompatible assumptions directly specified by the constraints, but removing any that are supersets of any other such nogood. For simplicity, in what follows we suppose each variable has the same set of domain values $1, \ldots, b$ so there are μb assumptions.

The general lattice representation of the problem consists of all possible sets of assumptions. However, for these CSPs, we need only consider those sets of assumptions in which each variable appears at most once, i.e., variables are not given more than one assignment. We thus restrict our discussion to a reduced lattice containing only these restricted sets. The sets can be arranged by size, i.e., the number of assumptions included in them. The number of sets or states of size k is given by $\binom{\mu}{k} b^k$, which counts the number of ways to choose the variables to appear in the set and the number of distinct assignments for those variables. If a certain set of assumptions is nogood according to some constraint, then all of its supersets will also be nogood. This is conveniently represented by linking each set of assumptions to those sets containing a single additional assumption. The resulting lattice for the case of 4 assumptions, A, B, C and D, is shown in Fig. 2.1. We will often refer to sets of a given size as being at the level of the lattice corresponding to their size, e.g., the empty set is at level zero, singleton sets are at level 1, etc.

As a concrete example, we consider a simple CSP that corresponds to this lattice. This problem consists of $\mu = 2$ variables (v_1 and v_2) each of which can take on one of $b = 2$ values (1 or 2) and the single constraint that the two variables take on distinct values, i.e. $v_1 \neq v_2$. Hence there are $\mu b = 4$ assumptions: $v_1 = 1$, $v_1 = 2$, $v_2 = 1$, $v_2 = 2$ which we abbreviate as A, B, C, D respectively. The corresponding lattice of sets of assumptions is given in Fig. 2.1. What are the minimized nogoods for this CSP? First there are those due to the explicit constraint that the two variables have distinct values: $\{v_1 = 1, v_2 = 1\}$ and $\{v_1 = 2, v_2 = 2\}$ or {A, C} and {B, D}. In addition, there are nogoods implied by the requirement that a variable takes on a unique value so that any set giving multiple assignments to the same variable is necessarily nogood. In this case, the minimized necessary nogoods are just $\{v_1 = 1, v_1 = 2\}$ and $\{v_2 = 1, v_2 = 2\}$ or {A, B} and {C, D}. Referring to Fig. 2.1, we see that these four minimized nogoods form a Sperner system consisting of sets, in

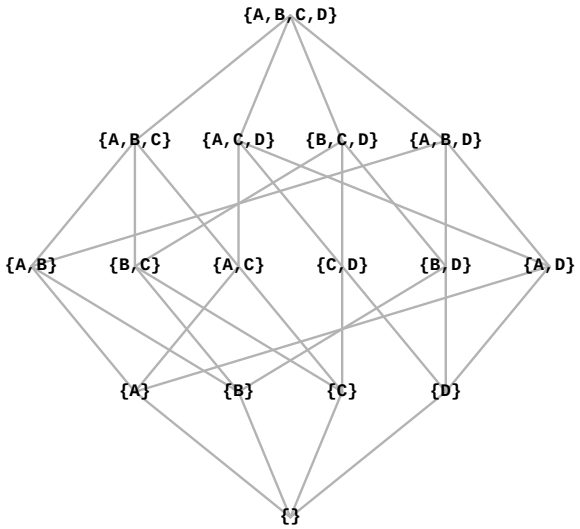

Figure 2.1: Structure of the power set of 4 assumptions. The subsets of {A,B,C,D} are grouped into levels by size. The bottom of the lattice, level 0, represents the single set of size zero, the four points at level 1 represent the four singleton subsets, etc.

this example, all of size two. These in turn force all sets of size 3 and 4 to be nogood too. However, sets of size zero and one are goods as are the remaining two sets of size two: {B, C} and {A, D} corresponding to $\{v_1 = 2, v_2 = 1\}$ and $\{v_1 = 1, v_2 = 2\}$ which are the solutions to this CSP.

We should note that these lattice structures correspond to the sets of assumptions that an ATMS creates [9]. Similar lattice and inheritance structures recur throughout A.I. systems [38] and, although the interpretation of the nodes and links may change, the basic structure remains the same.

2.2 Realistic Lattices

Each possible set of minimized nogoods gives rise to a corresponding search problem. However, to focus our analysis on cases that commonly arise in practice, in this section we examine some well known CSPs: graph coloring and satisfiability, both of which are NP-hard.

2.2.1 Graph Coloring

Graph coloring provides a good example since many practical constraint problems, e.g., scheduling, can be mapped into it [4]. The graph coloring problem consists of a graph, with μ nodes and an average connectivity[1] of γ, and a set of b colors. We denote this problem as b-COL. The task is to find an assignment of colors to nodes such that no two adjacent nodes (i.e., those at either end of an edge) have the same color. In the constraint formulation, the nodes correspond to variables which are assigned values corresponding to the colors.

The condition that no two adjacent nodes may have the same color means that the problem has a set of binary constraints, one for each edge in the graph. Such a constraint is violated when both nodes on the corresponding edge have the same color, which can happen in b ways. Thus each edge introduces b nogoods, and these are all distinct since different edges involve different nodes in the graph. Hence, the number of minimized nogoods arising from assigning the same color to linked nodes is

$$m = \frac{1}{2}\gamma\mu b \tag{2.1}$$

[1] i.e., the average number of edges incident on a node. Thus the total number of edges in the graph is $\frac{1}{2}\gamma\mu$ since each edge is incident on two nodes.

and all of these are of size $k = 2$. Finally, as any solution is an assignment of a color to every node, the solutions must be at level $L = \mu$ in the lattice.

2.2.2 Satisfiability

Another common constraint problem is satisfiability, or SAT, in which a propositional formula is given and one requires an assignment to the variables that make the formula true. As a specific instance [26], we consider k-SAT, in which the formula to be satisfied is a conjunction of clauses, each of which is a disjunction of exactly k variables (any of which may be negated).

In this problem, each of the μ variables appearing in the formula can take on one of two values, *true* or *false*. Thus in this case there are $b = 2$ values for each variable. Each clause appearing in the given formula is a disjunction of (possibly negated) variables. Hence the clause will fail to be true for exactly one assignment of values to the k variables appearing in it. This in turn gives rise to a single nogood, of size k. Distinct clauses will give rise to distinct nogoods, so the number of these nogoods is just the number of distinct clauses in the formula.

2.2.3 Other Cases

This discussion characterizes the lattice structure arising for some important CSPs. In particular for these cases the size of the minimized nogoods is small and fixed but their number and the size of the solutions both grow in proportion to the number of variables, μ. Beyond the specific examples considered above, we can expect that many realistic problems tend to have minimized nogoods of small size compared with the total number of assumptions. Intuitively this is reasonable for CSPs arising from interactions in the physical world which typically involve local interactions constraining a few variables at a time even when the number of variables is large. This suggests that there is a large range of problems with this scaling behavior. We adopt this behavior in the model discussed below, but note that other scaling behaviors can also be analyzed in the context of our theory.

Chapter 3

The Difficulty of Search

In this section we relate overall properties of the lattice structure to search cost, and then derive a connection between these properties and simple local characterizations of the nogoods specified by the constraints.

3.1 Global Measures

To illustrate the connection between structural properties of the lattice (or equivalently its minimized nogoods) and measures pertaining to the problem we consider the two most prominent aspects: the number of solutions and the cost to find those solutions (see Table 3.1). Other properties of interest are discussed briefly in §6.3. Although our "cost" measure will pertain to a search on the lattice we are not advocating that this is a good algorithm for doing an actual search. Instead we merely use the lattice as a way of describing the deep structure of constraint problems and our notional "cost" measure on it as an indicator of the difficulty of obtaining a solution using one of many possible algorithms.

parameter	interpretation
N_{soln}	number of solutions
$C(S)$	cost to find all solutions
$C_{\text{1st}}(S)$	cost to find a solution or determine there are none
$C_p(S)$	proxy for C_{1st}

Table 3.1: Global measures associated with a CSP. Here S denotes the set of minimized nogoods for the problem (a Sperner system). The quantitative values of the cost measures also depend on the particular search algorithm used. When this is not clear from context, we specify the search method as a superscript.

3.1.1 Number of Solutions

A solution to a CSP is an assignment of a value to each variable with all the constraints simultaneously satisfied. If there are μ variables and b values per variable, the total number of assumptions will be μb with all the solutions restricted to level $L = \mu$ in the lattice. The number of goods, $g(S, j)$, at level, j, is entirely determined by the minimized nogoods in the Sperner system S. In particular, the solutions are the goods at level L so the number of solutions is given by $N_{\text{soln}}(S) = g(S, L)$. No solutions can be above or below this level as all larger sets of assumptions must necessarily contain at least one conflicting variable assignment and all smaller ones do not contain a full complement of variables. The subsets of these solutions, i.e., the goods in the lattice, are called partial solutions since they correspond to a consistent assignment of values to some of the variables.

3.1.2 Cost

The cost of searching for a solution to a constraint problem can also be related to properties of the underlying set system. There are many different search methods, including those that involve modifying complete states until a solution is found or those, such as constraint preprocessing or abstraction, that attempt to restructure the problem. We briefly discuss these methods in §6, but here we focus on searches that involve incremental extension or combination of partial solutions until a complete solution is found, backtracking whenever it is impossible to extend a partial solution further. This includes a variety of search algorithms [32] that range from naive backtrack, with a random ordering for the variables and returning to the most recent decision point during backtrack, to sophisticated heuristic methods that attempt to select good variable orderings or backtrack more cleverly. These algorithms differ in their methods for avoiding redundant and irrelevant computation. Redundant computation can be avoided by caching past successes (directly or as generalizations) and reusing them as and when they subsequently arise [12, 23]. Similarly, irrelevant computation can be avoided by caching impossibilities and prohibiting further exploration of inferences dependent upon them [9, 10, 12].

The exact cost to solve a CSP depends on the search algorithm (e.g., simple backtracking) as well as what is required (e.g., any or all solutions). However, such information is too specific to give useful generalisations. Instead we choose to use a proxy for cost based on broad characteristics of many types of search processes. There are a number of plausible measures we could use: we focus here on cases applicable to searches that avoid most redundant and irrelevant search and that are also straightforward to express analytically within the framework of the Sperner set representation of constraint problems.

In our lattice representation of the search space, which represents all the partial solutions and the nogoods, we can implicitly model a clever search algorithm that caches all past successes (by generating each partial solution, i.e., good, no more than once) and that avoids all irrelevant computation (by skipping over nogoods which are ruled out by smaller nogoods). Specifically, an incremental search will continue expanding states as long as they are good, i.e., not pruned by one of the specified nogoods. Thus the number of goods and their size can serve as a natural proxy for the cost to search the entire space with any incremental search method. For the tree based backtracking searches, the variables will be instantiated in some specific order, so we can restrict our attention to those goods in the lattice that match that ordering. Backtracking methods that avoid redundant and irrelevant computation will examine goods of any size only once when searching for all solutions so a quantitative cost associated with Sperner system S is $C(S) = \sum_{j=0}^{L} g(S, j)$. The lower limit of the sum is somewhat arbitrary since different algorithms may start with different sized minimal configurations. Nevertheless, since, as we will see, most of the cost is due to a bulge in the number of goods relatively high in the lattice, this choice has no significant effect on our results. We should note that this measure is most directly applicable to cases in which there is no further domain-specific heuristics beyond the information available from the constraints themselves.

In many cases, a more interesting measure than the cost to find all solutions is the cost to find a single solution, or prove there are none. We can estimate this by the cost to find all solutions divided by the number of solutions, assuming the overall cost is roughly evenly divided among producing each solution, i.e., there is no particularly easy way to find the first solution, nor is it particularly difficult (as might indeed be the case if much initial effort is required to start the search before solutions are found, e.g., in breadth first search). Moreover, we assume that if there are no solutions, the full cost will be incurred before failure is recognized. Thus the cost proxy, C_p, we use is given by

$$C_p(S) = \begin{cases} C(S)/N_{\text{soln}}(S) & \text{if } N_{\text{soln}}(S) > 0 \\ C(S) & \text{if } N_{\text{soln}}(S) = 0 \end{cases} \tag{3.1}$$

which approximates the cost to find the first solution, if any, or else to determine there are no solutions.

3.2 Local Measures of Minimized Nogood Structure

As the structure of the overall lattice is completely determined by that of the minimized nogoods, we can learn everything about the lattice from studying just the minimized nogoods. However, a full specification of the set of minimized nogoods is too specific to give insight into the general behavior of searches. Instead, we need to step back and view the minimized nogoods at some level of abstraction. Our aim is to characterize

a set of minimized nogoods in sufficient detail to accurately determine interesting properties of the lattice they induce, but in general enough terms to allow simple models to predict these properties and be widely applicable.

Since the minimized nogoods prune states in the lattice, important characteristics relate to the amount of pruning they induce. The simplest properties are just the number of minimized nogoods and their size. Specifically, for a Sperner system S we define

- *Number of Minimized Nogoods*: $m = |S|$.
- *Average Size*: The average size of the sets in the Sperner system, $k = \frac{1}{m}\sum_{s\in S}|s|$.

As we will see, these measures are sufficient for predicting qualitative global properties and are even fairly accurate quantitatively. The use of additional properties is considered in §5 and 6.

3.3 Using Local Measures to Predict Global Behavior

To illustrate the connection between properties of the minimized nogoods and global aspects of the corresponding problem, we consider the class, or ensemble, of problems with μb assumptions and m minimized nogoods, all of which are the same size[1] k. In this case, any choice of the nogood subsets is guaranteed to be Sperner. Beyond this characterization of the problems, all their other properties are left unspecified and assumed to behave randomly. In particular, this means that any choice of the m nogoods is as likely as any other. There are $N_k = \binom{\mu}{k}b^k$ sets of size k in which each variable appears at most once, so that there are $\binom{N_k}{m}$ ways to select the nogoods and hence this many problems satisfying our specifications.

parameter	interpretation	scaling
μ	number of variables	$= \mu$
b	number of values	constant
k	size of the minimized nogoods	constant
L	level in the lattice of the solutions	$= \mu$
m	number of minimized nogoods	$\sim \beta\mu$ with constant β
N_k	number of sets of size k	$\sim (\mu^k/k!)b^k$

Table 3.2: Parameters used in our model and their assumed scaling behavior as the number of variables increases.

The cost measure and the number of solutions are determined by the number of goods in the lattice. For a node in the lattice to be a good at level $j \geq k$ none of its subsets of size k can be among the selected nogoods. Since this node is above exactly $\binom{j}{k}$ nodes at level k the number of ways to select m nogoods of size k such that a given node at level j is good is just

$$\binom{N_k - \binom{j}{k}}{m}$$

Since we suppose all choices of the nogoods are equally likely, the probability that this node is good p_j is then just the ratio of the number of choices in which it is good to the total number of choices for the nogoods, i.e.,

$$p_j = \frac{\binom{N_k - \binom{j}{k}}{m}}{\binom{N_k}{m}} \tag{3.2}$$

The expected number of goods of size j is just the total number of states at this level multiplied by the probability that such a state is good. There are $\binom{\mu}{j}b^j$ nodes at level j, but this counts all possible orderings

[1] We examine the consequences of relaxing this size restriction in §6. Moreover, a similar analysis shows the same general phenomena for large problems if we allow some variation in the number of nogoods among members of this class, e.g., by supposing each set at level k is selected to be nogood independently with a specified probability.

for instantiating the variables. With a single ordering, more relevant for backtrack searches, there are only b^j states so the expected number of goods of size j is

$$G(j) = b^j p_j \tag{3.3}$$

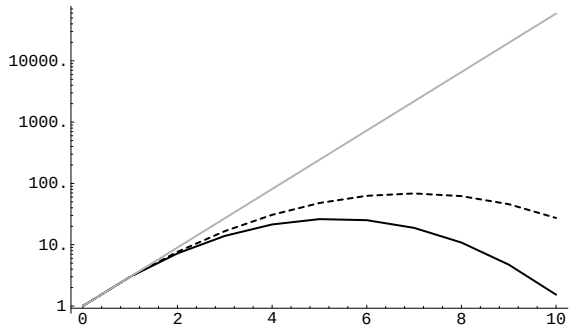

Figure 3.1: Total number of states and expected number of goods at various levels of the lattice for the case of $\mu = 10; b = 3; k = 3$ on a log scale. The gray curve shows the total number of states. The dashed and solid black curves show $G(j)$ vs. level j for the cases of $m = 60$ and $m = 80$ minimized nogoods, respectively. Note the bulge in the number of partial solutions at intermediate levels in the lattice contrasted against the monotonic rise in the size of the levels.

To provide a qualitative explanation of the behavior of this class of CSPs, Fig. 3.1 shows the behavior of the number of goods by level in the lattice. We see that the number of goods in a level initially grows rapidly, reaches a maximum at an intermediate level and then decreases before finally reaching the solution level. This is due to the competition between the growth in the total number of states at each level (the gray curve in the figure) and the pruning due to the minimized nogoods. Each minimized nogood prunes many more large sets than small ones, i.e., its pruning power increases with level in the lattice. The figure shows that the pruning power of the m minimized nogoods eventually overwhelms the growth in the number of states so that the number of goods begins to decrease after obtaining a maximum value at an intermediate level in the lattice. This bulge in the expected number of goods as a function of level in the lattice becomes increasingly sharp as μ increases and has been observed empirically [33].

We quantify these behaviors for large problems, i.e., as $\mu \to \infty$. The discussion of §2.2 suggests an interesting scaling limit is to keep k fixed and take $m = \beta\mu$, as summarized in Table 3.2. For behavior high in the lattice, with the scaling $j = \eta\mu$, approximating the binomial coefficients with Stirling's formula[2] gives

$$\ln p_j \sim \beta\mu \ln\left(1 - \left(\frac{\eta}{b}\right)^k\right) \tag{3.4}$$

which we use to obtain the behavior of the number of solutions and search cost.

3.3.1 Number of Solutions

As shown in Fig. 3.2, the expected number of solutions, $\langle N_{\text{soln}} \rangle = G(\mu)$ given by Eq. (3.3), grows exponentially with μ when there are sufficiently few nogoods and decays exponentially when there are many. This can be quantitatively understood by using Eq. (3.4) with $\eta = 1$ to get

$$\ln \langle N_{\text{soln}} \rangle \sim \mu \left(\ln b + \beta \ln\left(1 - b^{-k}\right)\right) \tag{3.5}$$

For small β, i.e., few nogoods, the first term in Eq. (3.5), which is positive, dominates and the expected number of solutions grows exponentially. Conversely, large β gives an exponentially small expected number of solutions, which means that most problems in this regime have no solutions. The transition between these

[2] I.e., $\ln(x!) \sim x \ln x - x$ for $x \to \infty$. This gives $\ln \binom{x-y}{z} - \ln \binom{x}{z} \sim z \ln(1 - \frac{y}{x})$ for $x - y \gg 1$ and $y \gg z \gg 1$.

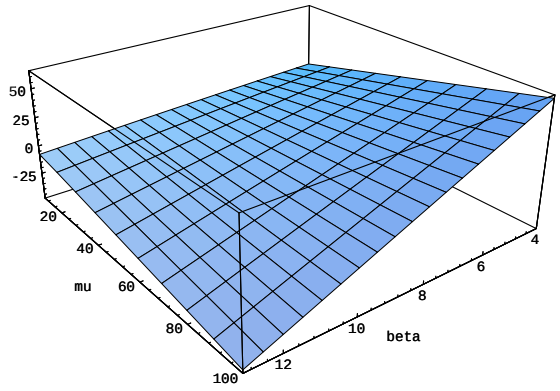

Figure 3.2: Behavior of $\ln \langle N_{\text{soln}} \rangle$ as a function of β and μ for the case in which ($b = 3; k = 2$. Note the growth in the number of solutions as μ increases when β is small, and the decrease when it is large. The linear growth or decay on this logarithmic scale corresponds to exponential growth or decay in the expected number of solutions as μ becomes large.

behaviors becomes increasingly sharp as μ grows, occurring at the critical value of β at which the expression for the leading order behavior of $\ln \langle N_{\text{soln}} \rangle$ equals zero, namely:

$$\beta_{\text{crit}} = \frac{\ln b}{-\ln\left(1 - b^{-k}\right)} \tag{3.6}$$

For example, $\beta_{\text{crit}} = 9.3$ for the case shown in Fig. 3.2. This transition in behavior is due to a competition between the increase in the number of states at the solution level as μ grows and the number pruned by the minimized nogoods. With a relatively small number of nogoods, the increase in the number of states dominates leading to a large number of solutions. Conversely, when there are many nogoods, most states are pruned and most such problems have no solutions. This corresponds to the behavior seen in Fig. 3.1.

3.3.2 Cost for All Solutions

The expected cost to find all solutions is $\langle C \rangle = \sum_{j=0}^{L} G(j)$,, the cumulative sum of the number of goods at all levels up to and including the solution level. Eq. (3.3) and (3.4) give

$$\ln G(j) \sim \mu \left(\eta \ln b + \beta \ln \left(1 - \left(\frac{\eta}{b} \right)^k \right) \right)$$

with $j = \eta\mu$. Thus as μ gets large, the sum for $\langle C \rangle$ will be increasingly dominated by those η values near $\hat{\eta}$, the value at which the expression for $\ln G(j)$ obtains its maximum value (in the range between 0 and 1). This maximum point can be specified implicitly by taking the derivative with respect to η and setting it to zero:

$$k\beta = f(\hat{\eta}) \equiv \hat{\eta} \left(\left(\frac{b}{\hat{\eta}} \right)^k - 1 \right) \ln b \tag{3.7}$$

which is valid when β is sufficiently large that the corresponding value satisfies $\hat{\eta} \leq 1$ (for smaller values, the maximum is at the solution level: this corresponds to a case in which there are very few nogoods and many solutions). As β increases, the corresponding value of $\hat{\eta}$ decreases, i.e., the maximum number of goods occurs at lower levels in the lattice.

Because of the increasing dominance of the largest terms in the sum, it can be approximated by an integral [1] whose leading behavior is simply

$$\ln \langle C \rangle \sim \mu \left(\hat{\eta} \ln b + \beta \ln \left(1 - \left(\frac{\hat{\eta}}{b} \right)^k \right) \right) \tag{3.8}$$

This decreases as more nogoods are added for two reasons: first, the maximum point moves lower in the lattice where there are fewer states (corresponding to a decrease in the first term), and, second, states at each level are increasingly pruned. This change in the size and location of the maximum (the bulge in the lattice) corresponds to the behavior illustrated in Fig. 3.1.

parameter	interpretation
$\langle N_{\rm soln}\rangle$	average number of solutions
$\langle C\rangle$	average cost to find all solutions
$\langle C_{\rm 1st}\rangle$	average cost to find a solution or determine there are none
$\langle C_p\rangle$	average value of the cost proxy for $C_{\rm 1st}$
$C_{\rm ap}$	approximate form for $\langle C_p\rangle$

Table 3.3: Average global measures associated with an ensemble of problems. The quantitative values of the cost measures also depend on the particular search algorithm used.

3.3.3 Cost to First Solution or to Failure

From the definition of Eq. (3.1) we have

$$\langle C_p\rangle = \left\langle \frac{C}{\max(1, N_{\rm soln})} \right\rangle \approx C_{\rm ap} \equiv \frac{\langle C\rangle}{\max(1, \langle N_{\rm soln}\rangle)} \tag{3.9}$$

This approximation, consisting of replacing the average of a function by the function of the average, assumes fluctuations are relatively small. It is commonly used to give a good qualitative idea of the behavior of statistical models and is known as the mean-field approximation. We evaluate its quantitative accuracy in §5.

We then have

$$\ln C_{\rm ap} \sim \begin{cases} \ln\langle C\rangle - \ln\langle N_{\rm soln}\rangle & \text{if } \langle N_{\rm soln}\rangle > 1 \\ \ln\langle C\rangle & \text{if } \langle N_{\rm soln}\rangle \le 1 \end{cases} \tag{3.10}$$

Hence, when $\langle N_{\rm soln}\rangle \le 1$, $\ln C_{\rm ap}$ decreases as nogoods are added since it is determined by the behavior of the total cost. With fewer nogoods, there is a competition between the decrease in the total cost and the even more rapid decrease in the expected number of solutions, given by Eq. (3.5), leading to a net increase in $\ln C_{\rm ap}$. This competition is also illustrated in Fig. 3.1 which shows that the expected number of goods at the solution level decreases far more rapidly than the number at the bulge. In summary then, the highest value of $C_{\rm ap}$ occurs at the point where $\langle N_{\rm soln}\rangle = 1$. Asymptotically, this transition point occurs at the critical value $\beta_{\rm crit}$ given in Eq. (3.6).

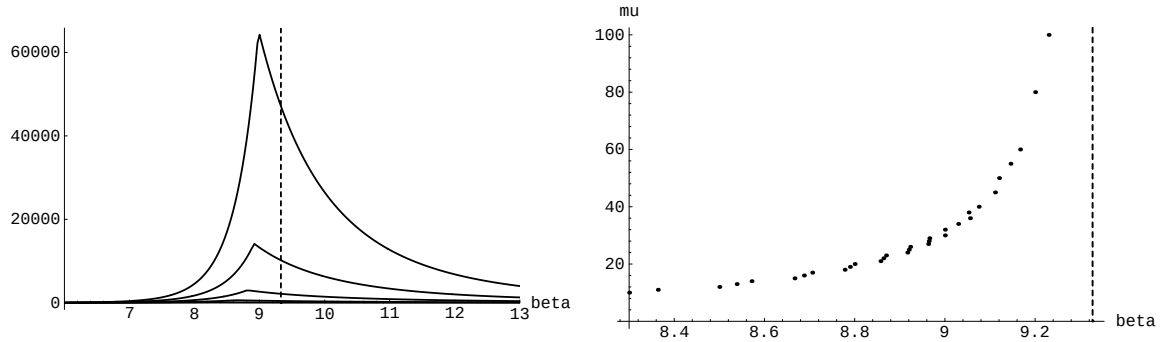

Figure 3.3: a) Behavior of $C_{\rm ap}$ as a function of β for $b = 3, k = 2$ for various values of μ, ranging from 10 to 30. Note the dramatic increase in cost at the critical value as well as the increasing sharpness of this effect as μ increases. The vertical line shows the asymptotic transition point at $\beta_{\rm crit} = 9.3$. b) Shift in the location of the peak in $C_{\rm ap}$ for larger μ, showing the convergence to the asymptotic value indicated by the dashed line.

The behavior of this cost estimate is shown in Fig. 3.3: there is a pronounced maximum for intermediate values of β, indicative of a phase transition, and this peak becomes increasingly sharp as $\mu \to \infty$. The figure also indicates how the peak achieves its asymptotic value, with it being somewhat below that value for small μ.

We therefore have a major result of this paper: as the number of minimized nogoods increases the cost per solution or to failure displays a dramatic rise at a critical number of minimized nogoods. Problems in this region are therefore significantly harder than others. This high cost is fundamentally due to the increasing pruning ability of minimized nogoods with level in the lattice: near the transition point the nogoods have greatly pruned states at the solution level (resulting in few solutions) but still leave many goods at lower levels near the bulge (resulting in many partial solutions and a relatively high cost).

3.3.4 Other Properties of the Transition Region

Our model can be used to investigate other properties associated with the transition region, such as the variance in the cost to find the first solution. As with the average value of our proxy, an explicit calculation of the variance is difficult. However, as the cost is dominated by a bulge in the number of goods at a single level, we might reasonably expect a crude approximation to the variance in our cost proxy is given by:

$$\mathrm{var}(C_p) \approx \frac{\mathrm{var}(C)}{\max(1, \langle N_{\mathrm{soln}} \rangle^2)} \tag{3.11}$$

As noted above, additional nogoods are much more effective at pruning states at the solution level than at the lower level containing the bulge which dominates the cost. This also applies to considering pairs of states involved in determining the variance in the number of goods. Hence, when $\langle N_{\mathrm{soln}} \rangle > 1$ the decrease in the number of solutions overwhelms the decrease in the variance in cost, leading to a net increase in this approximation. By contrast, when $\langle N_{\mathrm{soln}} \rangle < 1$ the denominator in Eq. (3.11) no longer changes so that the overall approximation decreases as nogoods are added. Thus, asymptotically, this approximation attains its maximum at β_{crit}, the phase transition point in the average cost. This suggests that the fluctuations in cost are indeed greatest at the transition, a property that is quite typical of phase transition phenomena. Incidentally, since our derivation for the maximum cost assumed that fluctuations were small, this indicates that the error introduced by this assumption is likely to be largest in the transition region. This is borne out in §5.

As a further application of our theory, we consider the behavior of the probability that a problem in our ensemble has a solution, P_{soln}, as the number of nogoods is increased. As with our cost proxy, this is difficult to determine exactly. However, we can gain some insight into its behavior with the same approximation as used in Eq. (3.9), namely that fluctuations are relatively small.

First, we note that, without approximation, the Markov inequality[3] implies that $P_{\mathrm{soln}} \equiv P(N_{\mathrm{soln}} \geq 1) \leq \langle N_{\mathrm{soln}} \rangle$ so when the average number of solutions goes to zero with increasing μ, P_{soln} must also go to zero. In particular, this means $P_{\mathrm{soln}} \to 0$ when $\beta > \beta_{\mathrm{crit}}$.

A corresponding lower bound, using the Chebyshev inequality[4], is

$$1 - P_{\mathrm{soln}} < \frac{\mathrm{var}(N_{\mathrm{soln}})}{(\langle N_{\mathrm{soln}} \rangle - 1)^2}$$

provided $\langle N_{\mathrm{soln}} \rangle > 1$. For $\beta < \beta_{\mathrm{crit}}$, $\langle N_{\mathrm{soln}} \rangle$ grows exponentially large with increasing μ and so this bound applies. If we now *assume* that fluctuations are relatively small, i.e., the ratio of the standard deviation in number of solutions to the average number of solutions goes to zero, we get $P_{\mathrm{soln}} \to 1$ for $\beta < \beta_{\mathrm{crit}}$.

Thus, within this approximation, we see that: 1) the probability to have a solution develops a sharp step with most problems having a solution below β_{crit} and almost none with solutions above this point, and 2) the location of this step is the at the location of the maximum cost. This observation is particularly interesting in that it provides a characterization of the transition point solely in terms of the problem structure (i.e., number of solutions) and independent of any particular search algorithm. This step can also be used to give an alternate derivation of the maximum in the cost proxy described above [40].

[3]For a nonnegative random variable X, $P(X > t) < \langle X \rangle / t$ [31].

[4]This is an application of the Markov inequality to $(X - \langle X \rangle)^2$ giving $P(|X - \langle X \rangle| \geq t) \leq \mathrm{var}(X)/t^2$. In our case, we need a bound on the probability to be significantly below the average: $P(X < \langle X \rangle - t) \leq P(|X - \langle X \rangle| \geq t) \leq \mathrm{var}(X)/t^2$.

In other words, our basic model not only predicts the qualitative existence of a phase transition in cost but also that the variance in cost should attain a maximum and the probability of a solution should exhibit a step *at exactly the same point* as the phase transition.

Chapter 4

Comparison of Theory to Actual Searches

To evaluate the accuracy of the above prediction of a phase transition in cost, we compare it with the behavior of actual constraint satisfaction problems.

4.1 Graph Coloring

For the graph coloring problem discussed in §2.2, Cheeseman et al. plot[1] the cost to find a solution, or determine none exist, as a function of the average connectivity of the graph γ. Empirically there is a sharp peak in the cost measure at some critical value of connectivity which we denote as $\gamma_{\text{crit}}^{\text{expt}}$. This coincides with an observed abrupt transition from near certainty of having solutions to near certainty of not having any. This behavior corresponds qualitatively to that predicted by our model.

# Nodes μ	# Colors b	β_{crit}	$\beta_{\text{crit}}^{\text{expt}}$
81	3	9.3	8.1 ± 0.3
144	3	9.3	8.1 ± 0.3
144	4	21.5	18 ± 1

Table 4.1: Comparison of our theory with Cheeseman et al.'s data for three graph coloring problems. Column three gives the predicted transition point from Eq. (3.6), recalling that $k = 2$ for graph coloring. The last column gives the corresponding measured values and rough estimates of their accuracy based on the plots of the data. These are related to the connectivities reported in [4] by Eq. (2.1), i.e., $\beta_{\text{crit}}^{\text{expt}} = \frac{1}{2} b \gamma_{\text{crit}}^{\text{expt}}$.

The quantitative test of our model for a number of graph coloring problems is given in Table 4.1. Our model predicts the qualitative phenomenon of a maximum in the search cost and, in addition, estimates the quantitative value of the phase transition point to within about 15%. Scaling is even better: as b changes from 3 to 4 this model predicts the transition point increases by a factor of 1.73, compared to 1.70 for the experimental data, a 2% difference. While these three experiments provide only a limited test of the quantitative accuracy of this model, they do suggest that the model is able to give the correct quantitative behavior of the location of the phase transition as μ and b are varied.

The outstanding discrepancies between our predictions and the observations are likely to be due to a mixture of the approximate evaluation of the transition point in our model, the lack of explicit correlations among the choices of nogoods in our model, the fact that Cheeseman et al. used "reduced" graphs rather than random ones to eliminate trivial cases, the fact that their search algorithm was heuristic whereas our model assumed a random ordering of variable choices, and statistical error in the samples they obtained.

[1] Figures 3b and 3c of [4]

4.2 Satisfiability

A peak in search cost has also been observed [4, 26] in k-SAT described in §2.2. The problem can be characterized by the ratio of the number of clauses in the formula to satisfy to the number of variables in the problem. For the case in which the clauses are randomly selected, a transition point for 3-SAT is observed [26] at the point where this ratio is 4.3. For problems with many variables, formulas with a fixed ratio of clauses to variables are unlikely to have any duplicate clauses so that each clause corresponds to a unique nogood of size 3. In this case, the ratio corresponds directly to the value of β with the transition point given by Eq. (3.6). For the case of $b = 2, k = 3$ corresponding to 3-SAT, this gives $\beta_{\text{crit}} = 5.19$, about 20% larger than the observed value.

Chapter 5

Robustness of the Model

The comparisons of the previous section showed that our model correctly predicts the qualitative phenomena and comes fairly close quantitatively. Going beyond this requires understanding which aspects of our derivation contribute most to the remaining error.

Two distinct sources of error can be identified. First, we used various mathematical approximations to simplify the derivation. Second, the model itself differs in detail from constraint problems such as graph coloring. For the first case, we consider the following issues:

- *Cost Proxy.* We assume that the cost to find the first solution, or determine there are none, is given by our proxy measure: $C_{1\text{st}} \approx C_p$. This eliminated the need to consider a specific search method or determine the distribution of solutions in the lattice.
- *Fluctuations.* The approximation of Eq. (3.9) assumes statistical fluctuations in the cost and number of solutions are small.
- *Comparison with Mean Value.* Our analysis focuses on the mean value for the cost, whereas the empirical data showed typical or median values.

The second source of error is that our model ignored the detailed structure of specific problems and assumed a fairly simple search method. Thus we examine the following:

- *Simple search.* We assume that the total cost is given by the sum of goods in the lattice restricted to an ordering of the variables, and, in particular that this ordering does not depend on the choice of minimized nogoods. Sophisticated heuristic search methods can adjust the ordering based on the specific nature of the problem.
- *Randomness.* Our model specifies only the size and number of minimized nogoods. We assume that remaining detailed degrees of freedom behave randomly.
- *All problem instances.* The empirical data for graph coloring was obtained for graphs that were guaranteed to have a solution [4] (although it was also noted that a similar transition was seen when all graphs were included). By contrast, our model includes all instances.

In the remainder of this section we examine both how well the mathematical approximations hold in a simple case, and the effect of relaxing the structural assumptions of our model. The main contributions to the error are summarized at the end of the section.

5.1 The Effect of Mathematical Approximations

To evaluate the effect of our mathematical approximations, we investigate a case in which the unspecified problem structure is in fact random, as assumed by our model, and compare its behavior to the model predictions. Any difference must then be due to errors introduced in the simplified derivation, rather than important problem structure ignored by our model. A sample problem is constructed by selecting a given

number of nogoods randomly from among all possible ones of a given size, thus enforcing the assumption of random unspecified structure.

We use a simple depth-first backtrack method to find the solutions. In this backtrack method, partial states are extended with assignments to additional variables until the state is pruned by a nogood or until all variables have assignments (the latter corresponds to a solution). The cost measure associated with this backtrack method is just the number of nodes in the search tree examined during the search, including the root. Because this simple search does no lookahead, these nodes will include the children of all goods (below the solution level), some of which may be pruned when examined. In terms of the number of goods at various levels in the lattice, the cost for this simple backtrack algorithm is

$$\begin{aligned} C^{\mathrm{bt}}(S) &= 1 + b \sum_{j=0}^{L-1} g(S,j) \\ &= 1 + b(C^{\mathrm{goods}}(S) - N_{\mathrm{soln}}(S)) \end{aligned} \tag{5.1}$$

where $C^{\mathrm{goods}}(S)$ is the cost based on a method which only examines the goods and which we used for our derivation in §3.3. Note that both quantities measure the cost to find all solutions, but differ due to different particular algorithms used. Because the dominant contribution to the cost comes from the bulge in the number of goods, the cost for simple backtrack and our theoretical measure essentially just differ by a factor of b which has no affect on the location of the peak.

We focus on the validity of three assumptions used in our derivation:

1. for large problems, our cost proxy accurately describes the search cost, at least in the transition region,
2. the approximation of small fluctuations, and
3. the mean value correctly captures the behavior of the phase transition.

5.1.1 Mathematical Approximation: Cost to First Solution is C_p

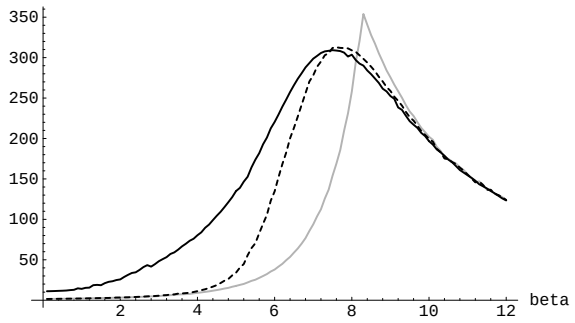

Figure 5.1: Search cost measures using simple depth-first backtrack for $\mu = 10, b = 3, k = 2$ (corresponding to 3-COL) vs. β, obtained by averaging over many samples. The solid curve shows the actual cost incurred $\langle C_{1\mathrm{st}} \rangle$ to find the first solution. The dashed curve is $\langle C_p \rangle$ with C_p for each problem instance given by Eq. (3.1). The gray curve is the approximation C_{ap} given by Eq. (3.9). The asymptotic predicted transition point is at $\beta_{\mathrm{crit}} = 9.3$. By comparison, the maximum in $\langle C_p \rangle$ is at 7.6 and the empirical data in Table 4.1 for 3-coloring corresponds to a transition point of 8.1. The small wiggles in the curves are statistical errors due to the limited number of samples taken at each value of β.

To examine the validity of our cost proxy, at least with respect to simple depth-first backtracking, Fig. 5.1 compares the cost measures we used. Comparing the black and dashed curves, we see that our cost proxy, $\langle C_p \rangle$, agrees well with the actual average cost to first solution, $\langle C_{1st} \rangle$, at and beyond the maximum. This means our proxy is correctly able to identify the existence and location of the phase transition, which is the most important consideration. For smaller values, i.e., when there are relatively few nogoods and hence many solutions, $\langle C_p \rangle$ underestimates the cost. This can be easily understood as a "start-up" cost incurred

by the backtrack method when finding the first solution that becomes relatively small when divided equally among all the solutions when computing C_p. For example, suppose there are no nogoods so all states are solutions and hence there is no pruning. Then depth-first backtrack will visit μ states, plus the root, as it proceeds from the root of the search tree to the first leaf so $C_{1\text{st}} = \mu + 1$. The cost to find all solutions will involve examining all nodes in the tree for a total cost of $(b^{\mu+1} - 1)/(b - 1)$ and will find all b^μ solutions. In this case we would have $C_p \sim b$ for large μ, considerably smaller than $C_{1\text{st}}$. Basically, the extra overhead of examining small partial states is divided among a very large number of solutions to give a relatively small value of C_p, whereas for the first solution, the overhead is applied entirely to the cost of finding one solution.

While this discussion has been based on simple backtracking, other search methods which incur a relatively large start-up cost before beginning to find solutions are also likely to have $C_{1\text{st}}$ much larger than our proxy C_p when there are many solutions. Nevertheless, as more nogoods are introduced and the number of solutions reduced, there are fewer solutions to divide any start-up overhead and we can expect that the cost proxy is closer to the true cost to first solution near and above the transition point.

From the samples used to obtain the data in Fig. 5.1 we can also examine the variance in the search cost for this ensemble of problems, and compare it to the variance of the proxy and its approximation given in Eq. (3.11). This is shown in Fig. 5.2. We note that the relation between the variance for simple backtrack and a method counting all the goods is obtained from Eq. (5.1):

$$\text{var}(C^{\text{bt}}) = b^2 \left(\text{var}(C^{\text{goods}}) + \text{var}(N_{\text{soln}}) - 2\text{cov}(C^{\text{goods}}, N_{\text{soln}})\right)$$

with the last term being the covariance of the cost and number of solutions.

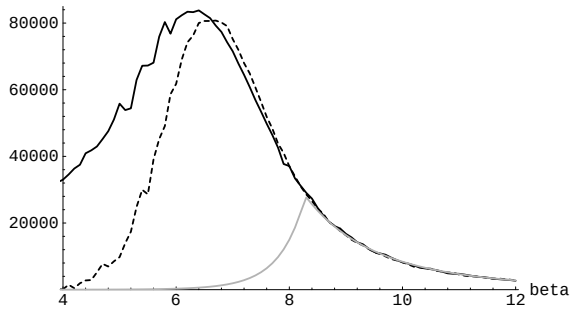

Figure 5.2: Variance in search costs for simple backtrack vs. β for $\mu = 10, b = 3, k = 2$ (3-COL). The solid curve shows $\text{var}(C_{1\text{st}})$, the dashed curve shows the variance in our proxy $\text{var}(C_p)$ and the gray curve shows the approximation to this given in Eq. (3.11).

5.1.2 Mathematical Approximation: Small Fluctuations

The second comparison in Fig. 5.1 is between the dashed and gray curves. This compares our cost proxy with its approximation used in Eq. (3.9). We see that this approximation underestimates the cost proxy below the transition point and overestimates it above, but is quite close away from the transition region. Consequently, this approximation overestimates the location of the transition point.

probability of solution

We also assumed fluctuations were small to derive the sharp step in the probability to have a solution. It is thus of interest to see how the large fluctuations, which cause the difference between the actual and approximate cost proxies shown in Fig. 5.1, affect this probability.

Fig. 5.3 shows how the probability to have a solution varies with the number of nogoods. As the size of the problem increases, this probability appears to be approaching a step, but at a value around $\beta = 7.4$, definitely below the critical value $\beta_{\text{crit}} = 9.3$ given by Eq. (3.6). This suggests that the probability to have a solution does indeed change abruptly at a critical value, but that this value is somewhat lower than predicted by our approximation, and in fact matches the behavior of the cost proxy as seen in Fig. 5.1. Thus there is a

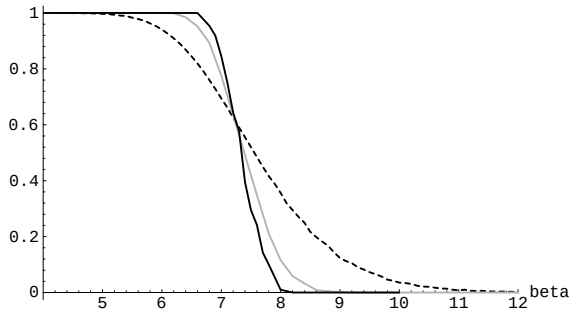

Figure 5.3: Probability to have a solution as a function of β for problems with $b = 3, k = 2$ and of various sizes: μ is 10 for the dashed curve, 50 for the gray one and 100 for the black one. In this case $\beta_{\text{crit}} = 9.3$. By comparison, the 50% probability point occurs at 7.5 for $\mu = 10$ (the case shown in Fig. 5.1) and slightly lower for larger μ.

regime, between the actual location of the step and β_{crit}, in which most problems have no solutions but the average number of solutions is nevertheless very large, i.e., the few problems with solutions have extremely many.

Theoretically, some bounds on this transition point can be obtained by relating this problem to the theory of random graphs. Specifically, the existence of solutions for a set of minimized nogoods of size 2 is equivalent to whether the largest set of nodes with no edges between them in the corresponding graph (the so-called independence number of the graph) has at least μ nodes. While it can be shown [2] that this is almost always true at least up to $\beta = 5.57$ and almost never true above $\beta_{\text{crit}} = 9.3$ for the example considered here, the question of precisely where the transition point is remains open.

corrections

Our estimate of the transition point can be improved through a perturbation series expansion. While this could be based on our cost proxy as given in Eq. (3.9), a simpler approach uses the observation that the transition coincides with the step in the probability for a solution. This latter quantity also has the advantage of depending only on characteristics of the problems (i.e., the number of solutions), not the particular search algorithm used. However, a perturbation expansion for P_{soln} is analytically complex. Instead we focus on the simpler quantity $f = \langle 1/(1 + N_{\text{soln}})\rangle$ which also depends only on the problem, not the algorithm used to solve it. Empirically, this quantity also exhibits an abrupt step at the same transition point, as shown in Fig. 5.4. This can be readily understood as follows. Above the transition point, $P_{\text{soln}} \to 0$ which implies that $f \to 1$. Below this point, we can expect most problems to have many solutions giving a small value for f. Thus we can identify the transition point as that value of β giving $f = \frac{1}{2}$.

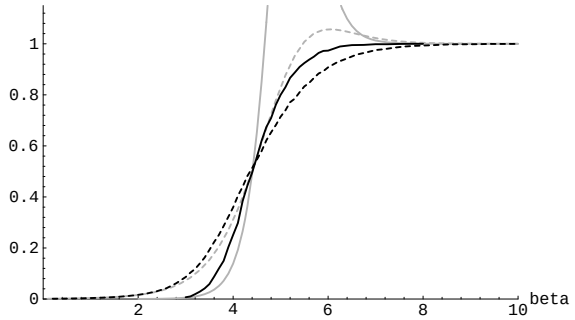

Figure 5.4: Behavior of $f = \langle 1/(1 + N_{\text{soln}})\rangle$ vs. β for $b = 2, k = 3$. The dark curves show empirical data for $\mu = 10$ (dashed) and $\mu = 20$ (solid). The light curves show the corresponding two-term Taylor series $F_1 + F_2$ approximations, which are quite accurate near the transition point of $f = \frac{1}{2}$.

We can obtain a perturbation expansion in terms of the size of the fluctuations [31] by writing

$$\begin{aligned} f(\epsilon) &= \left\langle \frac{1}{1+\langle N_{\rm soln}\rangle + \epsilon(N_{\rm soln} - \langle N_{\rm soln}\rangle)} \right\rangle \\ &= F_1 + \epsilon^2 F_2 + O(\epsilon^3) \end{aligned} \tag{5.2}$$

where

$$F_1 \equiv \frac{1}{1+\langle N_{\rm soln}\rangle},\ F_2 \equiv \frac{{\rm var}(N_{\rm soln})}{(1+\langle N_{\rm soln}\rangle)^3} \tag{5.3}$$

with $f = f(1)$. Note that while this series expansion converges for sufficiently small ϵ, for the value $\epsilon = 1$ of interest here the successive terms are not small. Nevertheless, this can be used to give a systematic perturbation expansion [1] because the terms, which involve successive moments of the distribution of solutions, are explicitly computable.

Using the first term F_1 the criterion $f = 1/2$ reduces to $\langle N_{\rm soln}\rangle = 1$ reproducing the previous result of Eq. (3.6) for $\beta_{\rm crit}$. Including the second term F_2 gives a smaller value $\beta^{(2)}_{\rm crit}$ at which $f = 1/2$, as shown in Fig. 5.4. In fact, for $\beta < \beta_{\rm crit}$, $\langle N_{\rm soln}\rangle \gg 1$ so the first term is exponentially small. Thus the criterion becomes simply $F_2 = 1/2$ or ${\rm var}(N_{\rm soln}) \sim \frac{1}{2}\langle N_{\rm soln}\rangle^3$. Since ${\rm var}(N_{\rm soln}) = \langle N^2_{\rm soln}\rangle - \langle N_{\rm soln}\rangle^2$, this criterion is equivalent to $\langle N^2_{\rm soln}\rangle \sim \frac{1}{2}\langle N_{\rm soln}\rangle^3$.

Using an argument similar to that used to derive the average number of solutions in §3.3, we have

$$\langle N^2_{\rm soln}\rangle = b^\mu \sum_{r=0}^{\mu} \binom{\mu}{r} (b-1)^{\mu-r} \frac{\binom{\binom{\mu}{k} b^k - 2\binom{\mu}{k} + \binom{r}{k}}{m}}{\binom{\binom{\mu}{k} b^k}{m}} \tag{5.4}$$

This is obtained by considering pairs of states at the solution level with r assignments in common and the probability that both members of the pair are goods. The asymptotic behavior of the terms T_ρ in this sum, with $r = \rho\mu$, is given by

$$\ln T_\rho \sim \mu \left(h(\rho) + (1-\rho)\ln(b-1) + \beta \ln\left(1 - \frac{2-\rho^k}{b^k}\right)\right) \tag{5.5}$$

where we have defined

$$h(x) = -x\ln x - (1-x)\ln(1-x) \tag{5.6}$$

which is positive (since x is between 0 and 1) with a single maximum at $x = 0.5$. The sum will be dominated by the value $\hat\rho(\beta)$ which maximizes $\ln T_\rho$. This can be given implicitly by setting the derivative of $\ln T_\rho$ to zero:

$$\beta = \frac{b^k - 2 + \hat\rho^k}{k\hat\rho^{k-1}} \ln \frac{\hat\rho(b-1)}{1-\hat\rho} \tag{5.7}$$

Combining these results, the transition point is determined by $\mu \ln b + \ln T_{\hat\rho} \sim 3\ln\langle N_{\rm soln}\rangle + O(1)$. Using the asymptotic behavior of $\langle N_{\rm soln}\rangle$ from Eq. (3.5) this criterion becomes

$$h(\hat\rho) + (1-\hat\rho)\ln(b-1) - 2\ln b + \beta \ln \frac{(b^k - 2 + \hat\rho^k)b^{2k}}{(b^k-1)^3} = 0 \tag{5.8}$$

This relation, together with Eq. (5.7), determine the corrected critical value $\beta^2_{\rm crit}$. These transcendental equations do not give a general closed form solution, but, for given values of b and k, can be solved numerically. For example, when $b = 2, k = 3$ this gives $\beta^2_{\rm crit} = 4.5$ which is more accurate that the original value of 5.2 from Eq. (3.6).

Further improvements can be obtained systematically by considering higher order terms in Eq. (5.2) which will involve higher moments of the distribution of the number of solutions. In this context, converting the Taylor series expansion of Eq. (5.2) to use rational functions of ϵ can be used to give more rapid convergence [1], at least in the vicinity of the transition point, even though the individual terms become large.

5.1.3 Mathematical Approximation: Comparison to Mean

Our analysis focuses on predicting the behavior of the mean search cost. If the problem instances include a few cases that are extremely hard to search, then the mean value could be far larger than most cases of the search. By contrast, the empirical data we compared our predictions with used typical or median costs to identify the peak. This more accurately reflects the behavior of most problem instances and is not affected if some particularly slow search cases are terminated before completion.

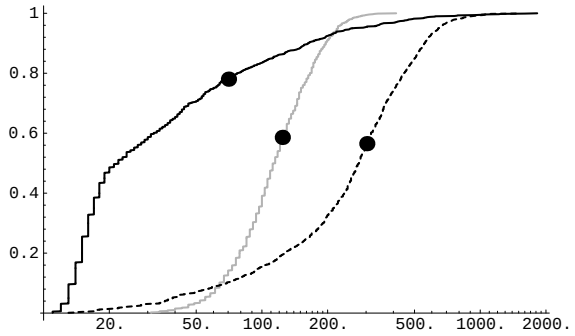

Figure 5.5: Cumulative distribution of cost $C_{1\text{st}}$ using simple backtrack for the case of $\mu = 10, b = 3, k = 2$ for various numbers of minimized nogoods: m is 40, 80 and 120 for the solid, dashed and gray curves respectively (or β values of 4, 8 and 12). These correspond to cases below, at and above the transition region. Specifically, each curve shows the fraction of 1000 search examples whose cost was less than or equal to the cost given with a log scale on the horizontal axis. The black dots show the location of the mean cost for each sample. The median cost corresponds to the point on each curve where the cumulative distribution equal 0.5.

This leads to the question of whether a few particularly difficult cases have an inordinate contribution to the mean search cost and even cause a shift in the peak value compared to the point at which the median cost is maximum. Fig. 5.5 shows the distribution of the actual search cost for three m values (below, at and above the transition region) for the case illustrated in Fig. 5.1. At and above the transition region we see that the distribution of cost values is fairly tight so that the mean gives a reasonable indication of the typical search cost. Below the transition region, however, the distribution develops a long tail indicating that a few very difficult cases raise the mean. This behavior, which becomes more extreme as μ increases, gives another contribution to the difference between our cost proxy and the actual cost observed below the transition in Fig. 5.1.

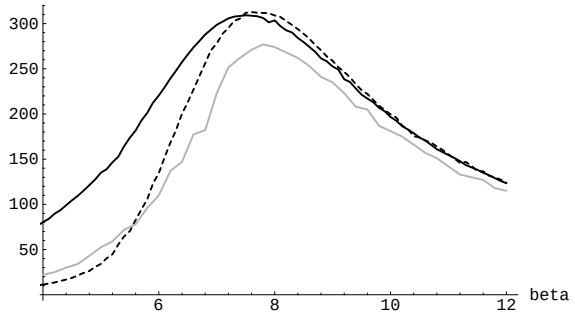

Figure 5.6: Comparison of mean (solid black curve) and median (gray) of $C_{1\text{st}}$ for simple backtrack with $\mu = 10, b = 3, k = 2$ as a function of β. For comparison, the cost proxy $\langle C_p \rangle$ is also shown (dashed).

The result of this long tail is shown in Fig. 5.6 which compares the median cost with the mean cost to first solution, and our proxy, shown previously in Fig. 5.1. We see that the median is closer to our proxy when there are few nogoods. This also changes the location of the maximum slightly to bring it into closer correspondence with that of our proxy measure. These changes become more significant as larger μ are considered, and in particular the location of the peak in our proxy is closer to that of the median cost than

that of the mean.

5.2 The Effect of Modelling Simplifications

5.2.1 Modelling Simplification: Simple Search

Our derivation for the expected total cost assumed that the order in which variables were examined did not depend on the particular nogoods. Sophisticated heuristics could base their choices on the nogoods and hence potentially change the transition point. Since the search methods used in the empirical evaluation of §4 employed heuristics, any such change could account for the discrepancy in our results.

To examine this possibility, we repeated our search on systems with unstructured nogoods, this time using a heuristic method. Specifically, instead of using a fixed ordering in which to instantiate the variables, at each step in the search we selected next that unassigned variable which was most constrained (i.e., had the fewest remaining consistent values). If there were several such variables, we selected that with the most unassigned variables appearing in nogoods that involved the variable. Any remaining ties were broken randomly. Note that if the most constrained unassigned variable had no possible values, then no extensions were possible and the search backtracked. In the case of graph coloring, this method corresponds to the Brelaz heuristic [20] used for the empirical data discussed in §4.

The measured search cost is the number of nodes examined. Because this heuristic only instantiates variables with at least one remaining consistent value, the nodes actually examined will always be goods in the lattice. Thus for any individual problem, this cost measure corresponds exactly with our identification of total cost with the sum of the number of goods in the lattice. However, because of the heuristic ordering of variables, the number of goods at each level will generally be substantially smaller than in our analysis which assumed a random ordering.

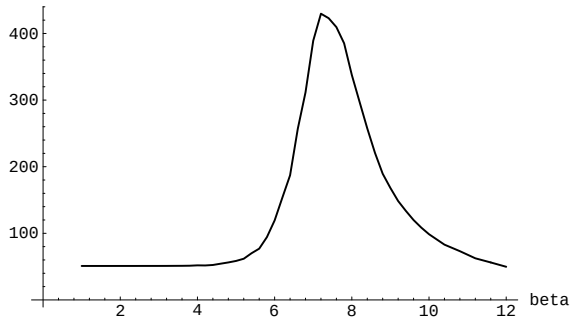

Figure 5.7: Mean cost to first solution $\langle C_{1\text{st}} \rangle$ using heuristic backtrack search for $\mu = 50, b = 3, k = 2$ as a function of β. The maximum occurs at $\beta = 7.2$. Note that the cost values are comparable to those shown in Fig. 5.1 but this case has five times a many variables, indicating the effectiveness of the heuristic.

An example of the behavior of this search is given in Fig. 5.7. Note that this method also has a peak and it is close to the location predicted by our proxy for random search. This suggests that the location of the peak in the cost identified by our proxy is not specific to simple search methods but instead captures an underlying structural change in the problem that is difficult for a wide range of backtrack methods.

As a further observation, this heuristic has a tighter distribution of costs than the random search so that the mean is a better approximation of the typical costs encountered. Nevertheless, the distribution when there are few nogoods still develops a long tail.

5.2.2 Modelling Simplification: Random Structure of the Nogoods

So far we have focused on families of problems in which only the number and size of the minimized nogoods is specified explicitly; all other structure is assumed to be random. In this section we examine the effects, on the phase transition, of specifying yet more structure explicitly. In particular we investigate what happens when the minimized nogoods have a higher or lower overlap than that obtained randomly and when there

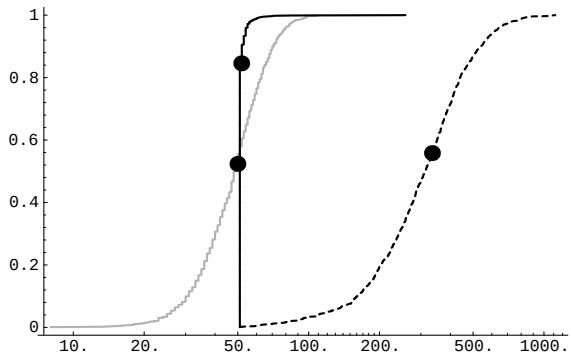

Figure 5.8: Cumulative distribution of heuristic search cost $C_{1\text{st}}$ for the case of $\mu = 50, b = 3, k = 2$ for various numbers of minimized nogoods: β is 4, 8 and 12 for the solid, dashed and gray curves respectively. Specifically, each curve shows the fraction of 1000 search examples whose cost was less than or equal to the cost given with a log scale on the horizontal axis. The black dots show the location of the mean cost for each sample.

are correlations between the values assigned to different variables. As we shall see, although the basic phase transition prevails, these additional degrees of freedom affect the location of the transition point.

pairwise overlap of the nogoods

To see how sensitive the phase transition phenomenon is to the precise pairwise overlap, we generated a range of systems with fixed number and size of nogoods but various overlaps and evaluated the cost proxy $\langle C_p \rangle$. Specifically, we constructed sets of size 2 drawn from a distribution in which those Sperner systems with average overlap

$$\theta = \frac{1}{m(m-1)} \sum_{s \neq s'} |s \cap s'|$$

occur with relative probability $e^{\alpha\theta}$ using the standard Metropolis algorithm [22]. Positive α values bias the selection toward Sperner systems with higher than random overlap, and conversely negative values bias the selection toward those with lower overlap. For comparison, when $\alpha = 0$ there is no bias and the selection corresponds to our previous model in which overlap was left unspecified.

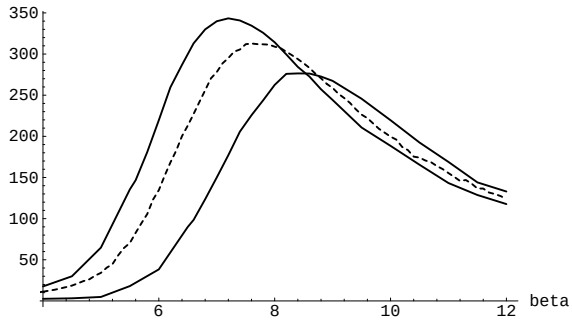

Figure 5.9: Cost $\langle C_p \rangle$ for simple backtrack vs. β for problems with $\mu = 10, b = 3, k = 2$ for various overlaps. From top to bottom, the data sets correspond to $\alpha = -300$, 0 and 300 respectively (the middle curve is the same as $\langle C_p \rangle$ shown as the dashed curve in Fig. 5.1). At the phase transition, the bottom set of points has an overlap one and a half standard deviations above random, and the upper set has an overlap one standard deviation below random. Far above the transition point, when there are many minimized nogoods, the overlap becomes extremely constrained and all cases have nearly the same overlap.

Fig. 5.9 shows the behavior of the cost proxy as a function of β for cases in which average overlap was specified to be below and above random, and, for comparison, the data from Fig. 5.1 in which the overlap was unspecified. We see that the basic phenomenon remains but the extra parameter introduces a shift in the transition point, specifically increasing the overlap moves the transition point to larger values. A similar

small shift is seen in the peak of the actual cost to first solution $\langle C_{1\text{st}} \rangle$, and it remains close to the peak in the proxy.

This illustrates how the use of additional order parameters can modify the quantitative predictions of the theory. For the graph coloring problem, the average overlap, which depends on both the average connectivity in the graph and its variance, is somewhat lower than the random case. However, this is a relatively small effect, and one which decreases as larger graphs are considered since the deviation from random overlap of the graph coloring nogoods decreases with μ.

independence of the nogoods

In the graph coloring example described in §2.2, an edge in the graph between nodes u and v introduces b minimized nogoods of the form $\{u = i, v = i\}$ for i from 1 to b. This is in contrast to our model which assumes all choices of the m nogoods are equally likely. We examine the effect of this structure by rederiving Eq. (3.3) to give the expected number of solutions for graph coloring.

Consider a state at the solution level, i.e., an assigned value for each of μ variables, in which the value i is used c_i times, with $\sum_i c_i = \mu$. In order for this state to be a solution, none of its subsets must be among the selected nogoods. This requires that the graph not contain an edge between any variables with the same assignment. This excludes a total of $\sum_i \binom{c_i}{2}$ edges. With random graphs with e edges, the probability that this given state will be a solution is just

$$p(\{c_i\}) = \frac{\binom{\binom{\mu}{2} - \sum_{i=1}^{b} \binom{c_i}{2}}{e}}{\binom{\binom{\mu}{2}}{e}} \tag{5.9}$$

By summing over all states at the solution level, this gives the expected number of solutions:

$$\langle N_{\text{soln}} \rangle = \sum_{c_1 \ldots c_b} \binom{\mu}{c_1 \ldots c_b} p(\{c_i\}) \tag{5.10}$$

where the multinomial coefficient counts the number of states with specified numbers of assigned values.

For the asymptotic behavior, note that the multinomial becomes sharply peaked around states with an equal number of each value, i.e., $c_i = \mu/b$. This also minimizes the number of excluded edges $\sum_i \binom{c_i}{2}$ giving a maximum in $p(\{c_i\})$ as well. Thus the sum for $\langle N_{\text{soln}} \rangle$ will be dominated by these states and Stirling's approximation can be used to give

$$\ln \langle N_{\text{soln}} \rangle \sim \mu \left[\ln b + \frac{\beta}{b} \ln \left(1 - \frac{1}{b} \right) \right] \tag{5.11}$$

because the number of minimized nogoods is related to the number of edges by $m = \beta\mu = eb$.

With this replacement for Eq. (3.5) our derivation of the transition point proceeds as before. Specifically, assuming small fluctuations gives the cost maximum at the point where the leading term of this asymptotic behavior is zero, i.e., at

$$\beta_{\text{crit}} = -\frac{b \ln b}{\ln(1 - 1/b)} \tag{5.12}$$

This result can also be obtained more directly by assuming conditional independence among the nogoods introduced by each edge [5]. For the cases of 3 and 4-coloring, this gives $\beta_{\text{crit}} = 8.1$ and 19.3, respectively, close to the empirical values given in Table 4.1.

5.2.3 Modelling Simplification: Ensemble with Specified Solution

In empirical studies of search methods, including the empirical data on graph coloring [4], one often restricts consideration to problems that are known to have a solution. One way to investigate these cases theoretically would be to use the previous theory but restrict consideration in the averages to only those problem instances with a solution. A simpler method, both theoretically and practically when generating such problems, is to use a prespecified state as a solution[1] and not allow any nogoods that are subsets of this specified solution.

[1] This is not the same as restricting the previous case to problems with solutions: a prespecified solution will cause problems with many solutions to be selected more frequently.

The specified solution introduces a few changes into the quantities describing the ensemble of problems. It changes the number of possible sets to $N_k = \binom{\mu}{k}(b^k - 1)$, of which m are selected to be the nogoods. There are $\binom{N_k}{m}$ ways to select these minimized nogoods and the relevant scaling remains $m \sim \beta\mu$.

Determining whether a given state is good must now take into account how many assignments, r, it shares with the specified solution. States that share many assignments with this solution are less likely to be pruned by a nogood. Specifically, for a node in the reduced lattice (i.e., the lattice consisting only of states with distinct variables) to be a good at level $j \geq k$ none of its subsets of size k can be among the selected nogoods. This node is above exactly $\binom{j}{k}$ nodes at level k. However, if this node has r assignments in common with the specified state, then $\binom{r}{k}$ of these excluded sets would never be picked anyway because they are subsets of the specified solution. Thus the number of ways to select m nogoods of size k such that a given node at level j is good is just

$$\binom{N_k - \binom{j}{k} + \binom{r}{k}}{m}$$

The probability that this node is good is then just the ratio of the number of choices in which it is good to the total number of choices for the nogoods, i.e.,

$$p_{jr} = \frac{\binom{N-k-\binom{j}{k}+\binom{r}{k}}{m}}{\binom{N_k}{m}} \tag{5.13}$$

There are $\binom{\mu}{j}b^j$ nodes at level j. However, during a tree search, the variables are instantiated in a particular order so the states that could be examined at a given level all have a single variable ordering. Thus the binomial factor can be dropped to get a cost measure that more closely corresponds to actual searches. In this case the expected number of goods of size j is

$$G(j) = \sum_{r=0}^{j} \binom{j}{r} (b-1)^{j-r} p_{jr} \tag{5.14}$$

where the other factors in the sum count the number of states at level j that have exactly r assignments in common with the specified solution. The expected number of solutions is $\langle N_{\text{soln}} \rangle = G(\mu)$.

For the asymptotic behavior, with $j = \eta\mu$ and $r = \rho j$, we have

$$\begin{aligned} \ln \binom{j}{r}(b-1)^{j-r} &\sim \mu\eta f_1(\rho) \equiv \eta\mu(h(\rho) + (1-\rho)\ln(b-1)) \\ \ln \rho_{jr} &\sim \mu\beta f_2(\rho,\eta) \equiv \beta\mu \ln\left(1 - \frac{\eta^k(1-\rho^k)}{b^k - 1}\right) \end{aligned} \tag{5.15}$$

with $h(\rho)$ given by Eq. (5.6). For each level, the sum giving the expected number of goods will be dominated by the term at which the sum of these expressions is maximum, which we denote as $\hat{\rho}(\eta, \beta)$.

With this behavior, one can proceed as before to determine a mean-field approximation for the transition point. Such an approximation is somewhat less accurate in this problem ensemble. However, our main focus here is to see whether the restriction to cases with solutions significantly changes the actual transition point identified by the cost proxy. Specifically, we see that the maximum in the cost proxy occurs at somewhat larger values of β than before. However, the peak in the median search cost is remains about the same as the previous ensemble, suggesting that our proxy for this ensemble is less precise in determining the transition point. Interestingly, this suggests that the results of our basic ensemble remain useful for predicting the location of the transition point, based on median cost, for this ensemble as well.

5.3 Summary

What can we conclude from the preceding experiments?

1. *Modelling approximations*: For certain problems it is important to explicitly account for the special correlations between the values assigned to different variables if one is to make quantitatively accurate predictions of the location of the phase transition point. However, we have shown by example that this is quite straightforward.

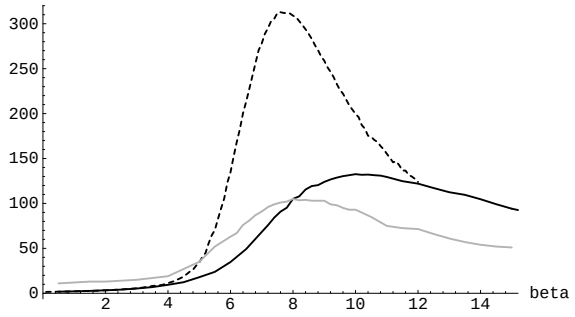

Figure 5.10: Comparison of cost proxy $\langle C_p \rangle$ of simple backtrack for the ensemble with specified solution (solid curve) and our basic ensemble (dashed). Parameters are $\mu = 10, b = 3, k = 2$, used in Fig. 5.1. The gray curve gives the median of the actual cost, $C_{\rm 1st}$, when there is a specified solution. Note that the cost values are significantly smaller when a solution is specified (i.e., the problems tend to be easier), but the peak in the median cost remains close to that of our basic ensemble.

2. *Mathematical approximations*: Of all the mathematical approximations we have made, the one that introduces the greatest error is the mean-field approximation (i.e. the assumption that fluctuations about the mean of the cost proxy are small). However, we have shown by example that inclusion of just the first order correction to our basic model allows it to yield quantitatively accurate predictions of the location of the phase transition point.

Our results are summarized in Table 5.1.

problem	expt	basic theory	basic+correlations	basic+corrections
3-COL	8.1 ± 0.3	9.3	8.1	N/A
4-COL	18 ± 1	21.5	19.3	N/A
3-SAT	4.3	5.2	N/A	4.5

Table 5.1: Comparisons of our basic theory and various refinements thereof with empirical data on graph coloring and 3-SAT discussed in §4.

Chapter 6

Extensions

6.1 Alternative Constraint Models

Our general approach can be used to predict the transition point for other ensembles of constraint problems. This approach consists of replacing Eq. (3.5) with the expected number of solutions for the new ensemble and then proceeding as before to find the transition point. Thus, our analysis can be readily applied to any constraint model for which the expected number of solutions can be computed. In this section we illustrate this for some common cases.

6.1.1 A Variable Number of Nogoods

One simple alternative to our basic model is to consider a class of problems which have somewhat different numbers of nogoods. Thus, instead of specifying that there are exactly m nogoods in each problem, we can suppose that each of the $N_k = \binom{\mu}{k} b^k$ possible assignments to k variables appears with probability ρ. The expected number of nogoods is then $\langle m \rangle = N_k \rho \equiv \beta\mu$, where we have selected the same scaling as used in our basic model.

In order for a given state at the solution level to be a solution, it must not have any of the selected minimized nogoods as a subset. Since there are $\binom{\mu}{k}$ such subsets, the probability to be a solution is

$$p_\mu = (1-p)^{\binom{\mu}{k}} \tag{6.1}$$

with the expected number of solutions being $\langle N_{\rm soln} \rangle = b^\mu p_\mu$. We then proceed as before to derive the transition:

$$\beta_{\rm crit} = b^k \ln b \tag{6.2}$$

which is a bit larger than Eq. (3.6), being closer when b^k is large.

6.1.2 A Two-Parameter Constraint System

Our basic model ignores any structure the set of nogoods may have. As we saw in the discussion of graph coloring leading to Eq. (5.12), including more specific structure can modify the predicted transition point. To see this more generally, we consider the structure imposed on the nogoods by the fact that they correspond to a set of constraints. In particular, we suppose that instead of simply having m randomly selected nogoods, we have a randomly selected constraints (each consisting of a distinct set of k variables) and for each constraint, n of its b^k possible assignments are selected to be locally nogood (i.e., to violate the constraint). In this case the total number of nogoods is $m = an$, and for each constraint, the conflicting assignments can be selected in $\binom{b^k}{n}$ ways.

Consider a state at the solution level. To be a solution, its assignments to each constraint's variables must not violate the constraint. Thus the probability for this state to be a solution is just

$$p_\mu = \frac{\binom{b^k-1}{n}^a}{\binom{b^k}{n}^a} = \left(1 - \frac{n}{b^k}\right)^a \tag{6.3}$$

with the expected number of solutions being $\langle N_{\rm soln}\rangle = b^\mu p_\mu$. We then proceed as before to derive the transition:

$$\beta_{\rm crit} = -\frac{n \ln b}{\ln(1 - nb^{-k})} \tag{6.4}$$

where we have used $\beta = m/\mu = an/\mu$.

Note this is the same as the basic model, Eq. (3.6), when $n = 1$ (as is the case for k-SAT since each clause is false for only a single assignment of its variables). However, for graph coloring, each edge corresponds to a constraint (so the parameter a is just the number of edges in the graph), and each constraint has b assignments that violate it (so $n = b$). Thus instead of reproducing our basic model, for graph coloring Eq. (6.4) is the same as Eq. (5.12) since $k = 2$ in this case. Thus this model, although not incorporating the complete structure of the nogoods for graph coloring, is sufficiently detailed to give the same result.

6.1.3 A Variable Number and Strictness of Constraints

The two-parameter model presented above may seem unrealistic in that it requires that all the constraints are equally strict, i.e., have the same number of local violations. This is the case for problems such as k-SAT or graph coloring, but won't hold in general. Thus we are led to consider a probabilistic version of this model in which we suppose each constraint exists with probability ρ_1 and each local assignment is chosen to be nogood with probability ρ_2. The expected number of nogoods is then $\langle m\rangle = \binom{\mu}{k}\rho_1 b^k \rho_2 \equiv \beta\mu$, where we have selected the same scaling as used in our basic model.

The probability that a given state at the solution level satisfies a particular constraint is $1 - \rho_2$, the probability that its assignments to the constraint's variables are ok. Thus the probability it satisfies all the constraints is

$$\begin{aligned} p_\mu &= \sum_{a=0}^{\binom{\mu}{k}} \binom{\binom{\mu}{k}}{a} \rho_1^a (1-\rho_1)^{\binom{\mu}{k}-a} (1-\rho_2)^a && (6.5)\\ &= (1 - \rho_1\rho_2)^{\binom{\mu}{k}} && (6.6) \end{aligned}$$

From this we obtain the transition point as

$$\beta_{\rm crit} = b^k \ln b \tag{6.7}$$

which is the same as Eq. (6.2), i.e., in this scaling limit the additional constraint structure has no affect on the transition point when variable numbers and sizes of constraints are allowed.

6.1.4 Minimized Nogoods of Different Sizes

Our basic model assumed that all the minimized nogoods were the same size. For many problems, e.g. graph coloring, this is correct. However, for other problems, such as generalized SAT [26], the minimized nogoods can be of different sizes. In this section we therefore extend our basic model to cover CSPs having minimized nogoods of different sizes.

As before we assume the constraint problem is over μ variables, each of which can take on one of b values. However, instead of all the nogoods being exactly the same size, we assume there are m_i of size i for $0 < i \le \mu$ and $m = \sum_i m_i$ is the total number of these nogoods. We let k be the average size of these nogoods and $\sigma(k)$ the deviation in their size. For example, having 1 nogood of size 2, 6 of size 3 and 1 of size 4 would result in $k = 3$ and $\sigma(k) = 1/2$. To define an ensemble of multilevel problems, we suppose the m_i nogoods at each level are selected randomly. While this could result in some selected sets being subsets of others, in our scaling limit $m = \beta\mu$ there are relatively few nogoods selected at each level provided we exclude nogoods of size 0 and 1, i.e., set $m_1 = m_0 = 0$, and not allow many very large nogoods. Hence this selection process is very likely to give Sperner systems as $\mu \to \infty$.

The derivation of §3.3 is readily modified for this multilevel case. Specifically, the probability that a set of j assumptions is good with respect to the m_i nogoods of size i is given by

$$p_j(i) = \frac{\binom{\binom{\mu}{i}b^i - \binom{j}{i}}{m_i}}{\binom{\binom{\mu}{i}b^i}{m_i}} \tag{6.8}$$

Hence the probability that a set of j assumptions is good with respect to the nogoods of *all* smaller sizes is simply:

$$\bar{p}_j = \prod_{i=1}^{j} p_j(i) \tag{6.9}$$

because the choices at each level are made independently. Thus Eq. (3.3) for the expected number of goods at level j generalizes to $G(j) = b^j \bar{p}_j$. This in turn gives the expected number of solutions, $\langle N_{\rm soln} \rangle = b^\mu \bar{p}_\mu$, and the expected cost to find all solutions $\langle C \rangle = \sum_j G(j)$. Using this in Eq. (3.9) for $C_{\rm ap}$ we obtain a search cost approximation for these multilevel problems.

As before, the transition point is where the leading order term of $\ln \langle N_{\rm soln} \rangle$ is zero. As our scaling limit we take $m_i = \beta_i \mu$ and note that $\binom{\mu}{i}(b^i - 1) \gg 1$ and $\binom{\mu}{i} \gg \beta_i \mu \gg 1$ since we restrict all nogoods to be at least of size 2 but relatively few of sizes $i \gg 1$. Moreover, we continue to suppose that the average size of the nogoods, k, remains constant. Eq. (3.5) then generalizes to

$$\ln \langle N_{\rm soln} \rangle \sim \mu \left[\ln b + \sum_{i=2}^{\mu} \beta_i \ln(1 - b^{-i}) \right] \equiv \mu \bar{\beta} \tag{6.10}$$

which defines a new generalized parameter $\bar{\beta}$. Thus the predicted phase transition point will always be located at $\bar{\beta}_{\rm crit} = 0$ regardless of the deviation in size of the minimized nogoods. This prediction is examined in Fig. 6.1 and 6.2. We have also examined larger problems and find that the peak does not move very much. As with our single level theory, the actual location of the phase transition is differs a bit from $\bar{\beta}_{\rm crit} = 0$ due to the assumption of small fluctuations in the mean-field approximation. Also, as for the single level case, our cost proxy is beneath the actual cost data in the region where there are exponentially many solutions. It is interesting to note that we have been able to find a single order parameter, sufficient to predict the location of the phase transition, that does not explicitly reference the average size or deviation in size of the minimized nogoods. Thus simple theoretical models can lead us to unexpected combinations of set system descriptors as composite order parameters.

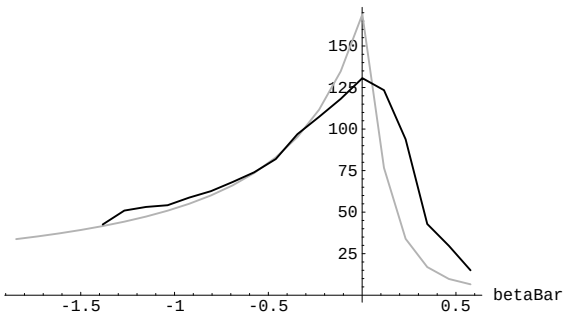

Figure 6.1: Cost measures vs. $\bar{\beta}$ for simple backtrack with $\mu = 10$, $b = 2$, $k = 3$ & $\sigma(k) = 0.5$ The solid curve is $\langle C_{\rm 1st} \rangle$ obtained empirically while the gray one is the theoretical approximation $C_{\rm ap}$.

As a check on our multilevel theory, we can compare its predictions against empirical data on generalized SAT problems reported by Mitchell et al. [26, p. 463]. Their data shows (in our terminology) that if one increases the deviation in size of the minimized nogoods, whilst keeping their average size fixed, the phase transition point occurs at a lower number of minimized nogoods and the value of the peak cost at the phase transition is diminished.

At first blush this might seem to contradict our prediction that the phase transition point occurs at a fixed value of $\bar{\beta}$. However, this is not so, as Mitchell et al. used the ratio of the number of clauses to the number of variables as their horizontal axis, corresponding to $m/\mu = \beta$, in our terminology, whereas we used an altogether different quantity, $\bar{\beta}$. Hence a good test of our model is to see whether, when we plot the theory curves in Fig. 6.1 and 6.2 against β rather than $\bar{\beta}$, we recover the wanderings of the phase transition they report. Such a plot, shown in Fig. 6.3, confirms that our multilevel theory does indeed capture the observed phenomenon. To understand what is happening we must realize that smaller nogoods prune much

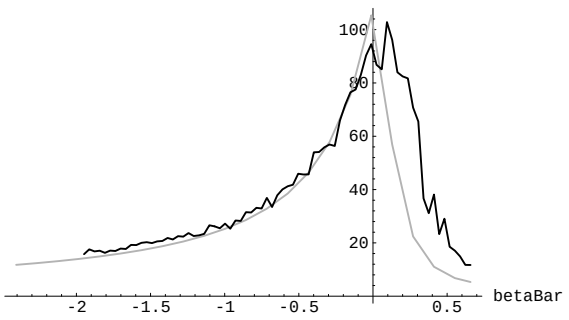

Figure 6.2: Cost measures vs. $\bar{\beta}$ for simple backtrack with $\mu = 10$, $b = 2$, $k = 3$ & $\sigma(k) = 1$. The solid curve is $\langle C_{\rm 1st} \rangle$ obtained empirically while the gray one is the theoretical approximation $C_{\rm ap}$.

more powerfully than larger ones. Thus if the average size of the nogoods is kept fixed, but the deviation in their size is increased, the set system as a whole becomes more and more like that of a set system at a single, lower, level.

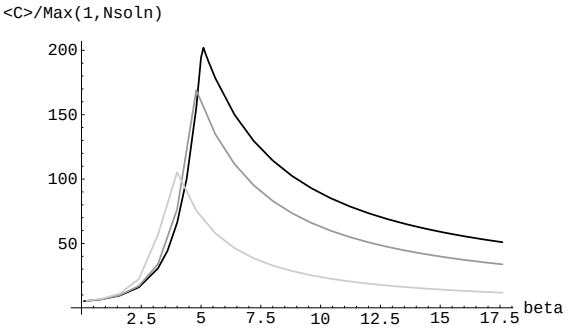

Figure 6.3: Theoretical approximation for the search cost, $C_{\rm ap}$, for simple backtrack, vs. β for problems with $k = 3$ exactly but $\sigma(k) = 0$ (darkest), 0.5 or 1 (lightest). Here $b = 2$ and $\mu = 10$.

6.2 Searches with Complete States

Our results, and the empirical observations, have been obtained in the context of a backtrack search. This raises the question of the applicability of these results to other search methods, in particular those that operate on complete states such as heuristic repair [24, 25], simulated annealing [21] and GSAT [36]. We saw that the cost for backtrack search was determined by the bulge in the number of goods, which occurs well below the solution level in the lattice. By contrast, it is not obvious how structural changes in the bulge will affect the search cost for methods that operate only with states at the solution level. In this section we investigate the behavior of such methods and show theoretically and empirically that they also have a region of increased cost at about the same point as the tree based searches.

Heuristic repair, simulated annealing and GSAT all attempt to improve a complete state through a series of incremental changes. These methods differ on the particular changes allowed and how decisions are made among them. In general they all guide the search toward promising regions of the search space by emphasizing local changes that decrease a cost function such as the number of remaining conflicting constraints. These heuristics provide useful guidance until a state is reached for which none of the local changes considered give any further reduction in cost. To the extent that many of these local *minimal* or *equilibrium* states are not solutions, they provide points where these search methods can get stuck. In such situations, practical implementations often restart the search from a new initial state, or perform a limited number of local changes that leave the cost unchanged in the hope of finding a better state before restarting. Thus the search cost for difficult problems will be dominated by the number of minimal points, $N_{\rm minimal}$,

encountered relative to the number of solutions, N_{soln}. Thus our proxy is:

$$\left\langle \frac{N_{\text{minimal}}}{N_{\text{soln}}} \right\rangle \approx \frac{\langle N_{\text{minimal}} \rangle}{\langle N_{\text{soln}} \rangle} \tag{6.11}$$

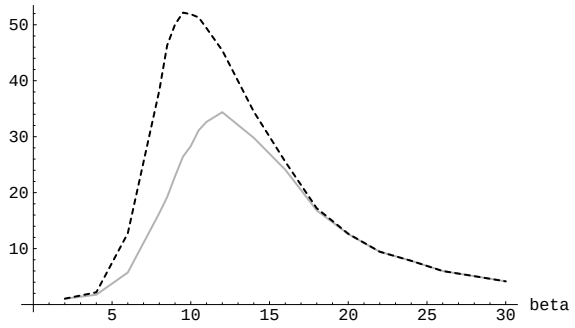

Figure 6.4: Ratio of number of minimal points to number of solutions vs. β for the case of $\mu = 10$, $b = 3$, $k = 2$ (dashed curve, with maximum at $\beta = 9.5$) and its mean-field approximation (grey, with maximum at 12).

This proxy has been examined, for example, in the specific context of the N-Queens problem [28]. The average number of minimal points can be determined by counting how many ways there are of choosing the minimized nogoods such that the neighbors of a given state at the solution level have at least as many conflicts as the state itself. The resulting expression is rather complicated and so we choose to simply plot it here (see Fig. 6.4). This predicts that the hardest problems occur around $\beta = 9.5$. Note, however, that the mean-field approximation again introduces some quantitative error. Compare this with empirical data, in Fig. 6.5, showing the cost of heuristic repair for increasing values of μ. We see that heuristic repair does indeed find certain problems harder than others and the numerical agreement between predicted and observed critical points is very good, suggesting that $\langle N_{\text{minimal}}/N_{\text{soln}} \rangle$ is an appropriate cost proxy. Thus our deep structure theory applies to sophisticated search methods beyond the tree search algorithms considered previously.

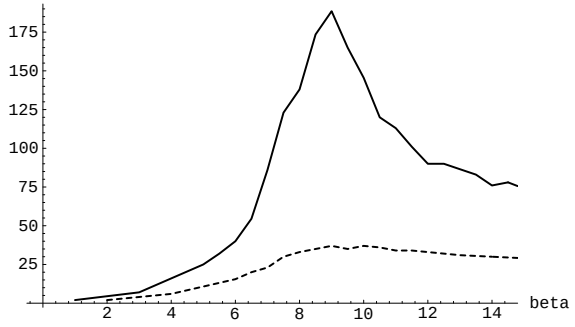

Figure 6.5: Median search cost for heuristic repair as a function of β for the case $b = 3$, $k = 2$. The solid curve, for $\mu = 20$, has a maximum at $\beta = 9$. The dashed curve, for $\mu = 10$, has a broad peak in the same region.

We should also note that this lattice structure can also be used to model other possible cost proxies for these search methods. One such example concerns the number of states around a local minimum with the same cost value. This, in effect, is the size of the local search space the method must navigate without heuristic guidance if it attempts to "escape" from a minimal state. If there are many such states, one can expect a random walk among them to take a long time to escape, hence giving a high overall search cost. Finally, there is the proxy based on a measure of the probability that each step will in fact reduce the search cost which has been examined, for example, in the context of heuristic repair [24]. This can be elaborated by considering the search as a random walk with various probabilities to reduce the number of conflicts at each step [29].

6.3 Additional Applications

In this paper we have used the lattice structure of CSPs to predict global properties such as the number of solutions and the search cost. As we outline in this section, there are many other ways of exploiting the lattice.

6.3.1 Potential for Benefit of Cooperative Methods

We have seen that the cost of solving problems depends on a few parameters describing the structure of the underlying minimized nogoods and is relatively large when there is an intermediate number of nogoods. A further question is whether there are general search methods that are particularly well suited for those problems that fall near the transition point. One possibility, noted in [4], is that the exceptionally large variance in search cost near the transition point suggests that using many independent versions of the search (but with different initial states or orderings in which variables are instantiated during backtrack) is likely to have at least one such search with cost significantly below the average. In a separate paper [18], we have built and tested an alternative approach that allows the different searches to exchange information in the form of partial solutions. In this way, the slower searches have the possibility to contribute some information to the faster ones. Our results suggest that such cooperative methods can greatly speed up problem solving and that they are most effective for the hardest problems near the phase transition [6, 18].

More generally, one can consider the benefit of additional domain-specific heuristics applied to the search. In particular, there is the question of how heuristics can be expected to interact with this range of problem difficulty. In the cases studied here the search cost grows exponentially with the size of the problem considered. When additional heuristics can be applied to prune nodes in a backtrack search earlier than is possible due to pruning by nogoods alone, there is the possibility of an abrupt transition from exponential to polynomial cost as the local pruning effectiveness of the heuristic is improved [19]. An interesting question is how the location of this transition may depend on the parameters describing the minimized nogoods. Addressing this issue requires some understanding of how the effectiveness of domain-dependent heuristics used for real problems can be expected to depend on the existence of large partial solutions. For example, cooperative methods can exploit these partial solutions to improve performance while other heuristics might have difficulty recognizing dead ends until close to the point where the nogoods would prune anyway.

6.3.2 Focussing Potential

Another property of lattice structure relating to the ease with which solutions might be found concerns how the solutions are distributed within their level. Intuitively, if the solutions are clustered together then certain subsets of assumptions must occur in solutions more frequently than one would expect from a random selection. Therefore, a problem solver might be able to spot these privileged subsets early on and use this information to bias its search. Similarly, the non-solutions must also contain other subsets more frequently than one would expect from random. Such "underprivileged" subsets could suggest which sets to avoid, affording a kind of negative focussing. Therefore, a tightly clustered solution distribution suggests there is potential for exploiting a clever focussing mechanism. In particular, by relating the clustering in solutions to appropriate structural properties of the minimized nogoods, this framework could provide estimates of how well focussing could be exploited from readily computed properties of the minimized nogoods.

6.3.3 Other Kinds of Problems

The analysis presented in this paper can easily be extended to problems other than constraint satisfaction. Here we briefly indicate a range of such problems.

optimizing and satisficing searches

In our formulation of constraint problems, each state was either consistent or inconsistent and the goal was to find a solution which satisfied all the constraints. Another important class of search problems consists in finding a state that is optimal with respect to some criterion. An important example of such problems is the traveling salesman problem in which one searches for the shortest tour of a specified set of cities. Empirical

observations show that this problem too exhibits a region of maximum search cost [4], suggesting that our analysis could be generalized.

The fact that the search cost for finding an exact solution in many search problems grows so rapidly forces one to also consider the question of how well one can do within a given resource bound. In these satisficing searches, the focus is on finding high quality, but not necessarily optimal, states rapidly. In a CSP context, the quality of a state could be characterized by how few constraints it violates [14] (perhaps weighted by a measure of each constraint's importance). More generally, where one is interested in a "high quality" solution rather than a "correct" solution one could associate a quality value with each partial solution instead of the boolean values "good" or "nogood". By defining a combination function which takes partial solutions together with their quality values and returns a larger partial solution and its quality value, one could explore the question of what happens to the quality of sets at the solution level as one varies the qualities of small partial solutions or the functional forms for the combination functions.

other lattice structures

Likewise, other kinds of lattice structured search spaces arise. For example, a classic approach to concept learning (the Version Space) exploits upper and lower hulls on a lattice as a compact representation for a space of possible concepts [16, 27]. In this interpretation the lattice nodes are variants of an emerging concept labelled as positive, negative or unseen instances of it. Structural properties of interest are, e.g., the size of the upper and lower hulls for a given number of concept exemplars. Such hulls are related to maximal and minimal surfaces on the lattice. Given some assumption about the structure of the concept to be learnt one could model the Version Space and ask how many examples are needed before the concept is probably known.

Another example is the lattice of assumptions used by the ATMS [9]. Here the assumptions are treated as unstructured and solutions can consist of various number of assumptions. Our analysis can also be applied to this case by considering general sets of assumptions.

The key point is that it should be possible to gain insight into the behavior of other kinds of problems by examining the lattices naturally induced in those cases.

Chapter 7

Related Work

In conjunction with the many empirical observations of the transition between easy and hard classes of problems, there have been a number of theoretical analyses of aspects of this phenomena. These include analyses based on particular CSP's (e.g., graph coloring and SAT [4, 5]), a number of exact bounds on the location of the transition point (e.g., for the chromatic number of graphs [2]), and exact evaluation for small-size problems such as SAT [8].

Regrettably, with the exception some insightful work by Provan [34], little attention has been paid to unravelling the properties of families of minimized nogoods. Provan's model differs from ours by using a more detailed specification of the minimized nogoods which included the number of sets by size together with their intra-level and inter-level overlaps. We contend that as A.I. systems scale up such details become progressively less important. In the limit of large problems, a few order parameters appear to be adequate predictors of behavior. Experience with modelling other kinds of A.I. systems [19, 41] leads us to believe this phenomenon is quite common: relatively small A.I. systems, for which the details most definitely do matter, do not always supply the right intuitions about large problems.

By introducing the underlying Sperner systems, we have extended the analysis of this phenomenon to a wide range of constraint satisfaction problems. By abstracting away from the detailed nature of individual problems, we have shown the general nature of the transition. As future work, this method of analysis could be applied to an additional range of problems whose underlying structure involves lattices of sets. As discussed above, these include some optimization problems other search methods. A better understanding of the lattice structure could also provide a basis for constructing more powerful search algorithms.

Chapter 8

Conclusions

We introduced the concept of the "deep structure" (or set of minimized nogoods) of a constraint satisfaction problem as that set system produced by collecting the nogood ground instances of each constraint and keeping only those that are not supersets of any other. We then showed how to use such deep structure to predict where, in a space of problem instances, the hardest problems are to be found, where fluctuations in difficulty are greatest and where the probability of having any solutions drops abruptly to zero. Interestingly, our theory predicts all these things occur at exactly the same place, confirming empirical observations made by other authors.

The remaining numerical discrepancy between our basic theory and the reported data led us to examine the assumptions underlying our theory. After extensive analysis, we identified the two principal sources of error as the assumption that the values assigned to different variables were uncorrelated and the assumption that the fluctuations about the mean of the cost proxy were small. Equipped with this knowledge, it was possible to find the simplest modifications that enabled quantitatively accurate estimates of the location of the phase transition points for each of the problem types considered, to be made. Furthermore, the perturbation procedure given by Eq. (5.2) demonstrates how to go beyond the "independence" assumptions one is usually forced to make in order to have a tractable theory, to achieve arbitrarily accurate predictions. This method might be of use to someone wanting to repeat our analyses on other problems in, perhaps, other scaling regimes. However, we should caution that our theory is not intended to be used to predict the difficulty of an individual CSP but rather to predict the average difficulty of CSPs drawn from some pool (or ensemble) or similar problems.

The exact form for the cost proxy used to predict difficulty does not appear to be that critical, provided it tracks the actual cost measure faithfully. In particular, backtrack algorithms on the lattice or in a tree, a heuristic tree search algorithm and a heuristic repair algorithm all encounter their most difficult problems in the same region. This allows us to finesse the need to do *algorithmic* complexity analyses by essentially doing a *problem* complexity analysis in terms of the underlying set of minimized nogoods.

The choice of ensemble is problematic in that ideally one should like to pick one that conforms to problems arising in the real world and scales up in a realistic fashion. Unfortunately, this is likely to vary from domain to domain. Consequently, the best we could do was to pick an ensemble that seemed to cover a wide variety of actual CSPs. This meant assuming that an instance in the ensemble had the characteristics reported in Table 3.2. Certain degrees of freedom in its specification need to be set explicitly, such as the number and average size of the nogoods, whereas others are treated as random variables, such as the extent to which the nogoods overlap. The explicit degrees of freedom are the order parameters against which we calculate the variation of problem difficulty. Consequently, for example, our basic model predicts how problem difficulty, of CSPs having nogoods of equal size, varies as the number of nogoods is increased. Results on refinements to this model to account for nogoods of mixed size and nogoods that overlapped more or less than that obtained randomly were also studied. In all cases, we found that the basic easy-hard-easy pattern prevailed suggesting that the phase transition phenomenon is quite generic and that the key order parameters are the number and size of the minimized nogoods. In the case of the multi-level version of our theory, the theoretical work lead us to conjecture a new composite order parameter whose form we could not have envisaged by guessing alone.

There is much work to be done understanding the mathematical properties of Sperner systems augmented with the variable/value structure. The current results on fully general Sperner systems [3] proved to be too weak to be of much use to us apart from a theorem that we used to test the possibility of having a Sperner system with prescribed numbers of sets of various sizes. We expect the extension of these theorems to accommodate the special nogood structure will lead to important results in combinatorics. Also, extending our model to applications other than locating regions of difficulty seem possible. However, it would appear that even with a rather simple model we can achieve considerable insight into the fundamental structure of constraint problems.

These observations bode well for the future utility of this kind of analysis applied to other A.I. systems.

Acknowledgments

We thank Mark Stefik for many productive discussions during the preparation of this paper.

Bibliography

[1] Carl M. Bender and Steven A. Orszag. *Advanced Mathematical Methods for Scientists and Engineers*. McGraw Hill, NY, 1978.

[2] B. Bollobas. *Random Graphs*. Academic Press, NY, 1985.

[3] Bela Bollobas. *Combinatorics*. Cambridge University Press, Cambridge, 1986.

[4] Peter Cheeseman, Bob Kanefsky, and William M. Taylor. Where the really hard problems are. In J. Mylopoulos and R. Reiter, editors, *Proceedings of IJCAI91*, pages 331–337, San Mateo, CA, 1991. Morgan Kaufmann.

[5] Peter Cheeseman, Bob Kanefsky, and William M. Taylor. Computational complexity and phase transitions. In *Proc. of the Workshop on Physics and Computation (PhysComp92)*, pages 63–68, Los Alamitos, CA, 1992. IEEE Computer Society Press.

[6] Scott H. Clearwater, Bernardo A. Huberman, and Tad Hogg. Cooperative problem solving. In B. Huberman, editor, *Computation: The Micro and the Macro View*, pages 33–70. World Scientific, Singapore, 1992.

[7] Paul R. Cohen, Michael L. Greenberg, David M. Hart, and Adele E. Howe. Trial by fire: Understanding the design requirements for agents in complex environments. *AI Magazine*, 10(3):33–48, Fall 1989.

[8] James M. Crawford and Larry D. Auton. Experimental results on the cross-over point in satisfiability problems. In Haym Hirsh et al., editors, *AAAI Spring Symposium on AI and NP-Hard Problems*, pages 22–28. AAAI, 1993.

[9] J. de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

[10] R. Dechter. Learning while searching in constraint-satisfaction problems. In *Proc. of AAAI-86 5th Natl. Conf. on AI, 11-15 Aug 1986*, pages 178–183, Los Altos, CA, 1986. Morgan Kaufmann.

[11] Johan deKleer. A comparison of ATMS and CSP techniques. In N. S. Sridharan, editor, *Proc. of the 11th Intl. Joint Conf. on AI (IJCAI89)*, San Mateo, CA, 1989. Morgan Kaufmann.

[12] C. Elkan. Conspiracy numbers and caching for searching and/or trees and theorem proving. In *Proceedings of IJCAI89*, pages 341–348, Los Altos, CA, 1989. Morgan Kaufmann.

[13] John Franco and Marvin Paull. Probabilistic analysis of the davis putnam procedure for solving the satisfiability problem. *Discrete Applied Mathematics*, 5:77–87, 1983.

[14] Eugene C. Freuder and Richard J. Wallace. Partial constraint satisfaction. *Artificial Intelligence*, 58:21–70, 1992.

[15] Craig Gotsman. A cluster detection algorithm based on percolation theory. *Pattern Recognition Letters*, 12:199–202, April 1991.

[16] Haym Hirsh. Polynomial-time learning with version spaces. In *Proc. of the 10th Natl. Conf. on Artificial Intelligence (AAAI92)*, pages 117–122, Menlo Park, CA, 1992. AAAI Press.

[17] Tad Hogg and J. O. Kephart. Phase transitions in high-dimensional pattern classification. *Computer Systems Science and Engineering*, 5(4):223–232, October 1990.

[18] Tad Hogg and Colin P. Williams. Solving the really hard problems with cooperative search. In Haym Hirsh et al., editors, *AAAI Spring Symposium on AI and NP-Hard Problems*, pages 78–84. AAAI, 1993.

[19] B. A. Huberman and T. Hogg. Phase transitions in artificial intelligence systems. *Artificial Intelligence*, 33:155–171, 1987.

[20] David S. Johnson, Cecilia R. Aragon, Lyle A. McGeoch, and Catherine Schevon. Optimization by simulated annealing: An experimental evaluation; part ii, graph coloring and number partitioning. *Operations Research*, 39(3):378–406, May-June 1991.

[21] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[22] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chemical Physics*, 6:1087–1092, 1953.

[23] S. Minton. Quantitative results concerning the utility of explanation-based learning. In *Proc. 7th National Conference on Artificial Intelligence*, pages 564–569, St. Paul, Minnesota, 1988.

[24] Steven Minton, Mark D. Johnston, Andrew B. Philips, and Philip Laird. Solving large-scale constraint satisfaction and scheduling problems using a heuristic repair method. In *Proceedings of AAAI-90*, pages 17–24, Menlo Park, CA, 1990. AAAI Press.

[25] Steven Minton, Mark D. Johnston, Andrew B. Philips, and Philip Laird. Minimizing conflicts: A heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence*, 58:161–205, 1992.

[26] David Mitchell, Bart Selman, and Hector Levesque. Hard and easy distributions of SAT problems. In *Proc. of the 10th Natl. Conf. on Artificial Intelligence (AAAI92)*, pages 459–465, Menlo Park, 1992. AAAI Press.

[27] T. M. Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

[28] Paul Morris. On the density of solutions in equilibrium points for the queens problem. In *Proc. of the 10th Natl. Conf. on Artificial Intelligence (AAAI92)*, pages 428–433, Menlo Park, CA, July 1992. AAAI Press.

[29] Ron Musick and Stuart Russell. How long will it take? In *Proc. of the 10th Natl. Conf. on Artificial Intelligence (AAAI92)*, pages 466–471, Menlo Park, 1992. AAAI Press.

[30] Christos H. Papadimitriou. On selecting a satisfying truth assignment. In *Proc. of 32nd Annual Symposium on Foundations of Computer Science*, pages 163–169, 1991.

[31] A. Papoulis. *Probability and Statistics*. Prentice Hall, Englewood Cliffs, NJ, 1990.

[32] J. Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, Reading, Mass, 1984.

[33] G. M. Provan. Efficiency analysis of multiple-context TMSs in scene representation. Technical Report OU-RRG-87-9, Robotics Research Group, Dept. of Engineering Science, Univ. of Oxford, 1987.

[34] Gregory M. Provan. Efficiency analysis of multiple-context TMSs in scene representation. In *Proceedings of AAAI87*, pages 173–177, Los Altos, CA, 1987. Morgan Kaufmann.

[35] Paul Walton Purdom, Jr. Search rearrangement backtracking and polynomial average time. *Aritificial Intelligence*, 21:117–133, 1983.

[36] Bart Selman, Hector Levesque, and David Mitchell. A new method for solving hard satisfiability problems. In *Proc. of the 10th Natl. Conf. on Artificial Intelligence (AAAI92)*, pages 440–446, Menlo Park, CA, 1992. AAAI Press.

[37] J. Shrager, T. Hogg, and B. A. Huberman. Observation of phase transitions in spreading activation networks. *Science*, 236:1092–1094, 1987.

[38] D. S. Touretzky. *The Mathematics of Inheritance Systems*. Morgan Kaufmann, Los Altos, CA, 1986.

[39] M. S. Waterman, L. Gordon, and R. Arratia. Phase transitions in sequence matches and nucleic acid structure. *Proc. Natl. Acad. Sci. USA*, 84:1239–1243, 1987.

[40] Colin P. Williams and Tad Hogg. Using deep structure to locate hard problems. In *Proc. of the 10th Natl. Conf. on Artificial Intelligence (AAAI92)*, pages 472–477, Menlo Park, CA, 1992. AAAI Press.

[41] Colin P. Williams and Tad Hogg. The typicality of phase transitions in search. *Computational Intelligence*, 9(3):221–238, 1993.