The Typicality of Phase Transitions in Search

Colin Williams and Tad Hogg

Dynamics of Computation Group
Xerox Palo Alto Research Center
Palo Alto, CA 94304

Abstract

Search is fundamental to Artificial Intelligence, and numerous sophisticated search methods have been developed. We present a general, simple model of search processes and use it to analytically determine some typical behavior when applied to large problems. In particular, this identifies abrupt changes in overall search cost as small improvements are made in the underlying method. We also examine the robustness of this model's predictions in a range of more realistic cases. More generally, we introduce a criterion for determining when average case results reflect typical behavior which allows the method developed here to be used for investigating other large-scale behaviors of complex AI systems.

Keywords: search, complexity, typicality, phase transitions
Xerox PARC SSL technical report 1992
A shorter version of this technical report appeared in *Computational Intelligence* **9** 221–238 (1993).

1. Introduction

Search is one of the most pervasive techniques in Artificial Intelligence. Problem solving, diagnosis, design and planning, for example, can all be cast as a search through some space of alternatives. Given its central importance to the field, considerable work has been devoted to developing a variety of search methods and determining the situations for which they are best suited [2, 4, 9, 13–15, 17].

Distinct search algorithms differ in the adroitness with which they navigate search spaces. This navigation can consist of examining a series of fully specified states until a desired one is found, or can be a process of incrementally constructing a solution by combining a series of intermediate states or partial solutions. Each time some search state is examined the algorithm must decide whether to pursue it or defer it until later (possibly forever) and move on elsewhere. Hence a simple characterization of a key part of a wide range of A.I. programs is as a search process in which particular choices are more or less likely to be preferred. The power of a search method is then determined by how well it discriminates between choices that lead quickly to solutions and those that do not.

The standard approach to analyzing search algorithms is to compute best, worst or average case complexities and exploit very detailed models of the underlying search process. Unfortunately, such predictions can be misleading because there is no guarantee the typical behaviour will mimic best, worst or average case behaviour. Moreover, an overabundance of detail can confuse the key mechanisms with the incidental ones and hinder true understanding of the search process. This situation can only get worse as progressively more complex search techniques are developed. What is needed is some unified way to predict the typical behaviour of general search processes.

This paper presents such an approach based on an application of branching processes [5, 7]. We develop a criterion of when predicted behaviour is typical and apply it to a general model of search that encompasses algorithms ranging from the naive to the sophisticated. Hence our model provides a very broad view of search processes and allows us to delimit the regimes in which certain universal

phenomena can be expected without incurring the penalty of a complicated model.

Two distinct types of search are possible; satisficing searches in which states have various quality values and one attempts to find relatively high value states, and exact searches in which there is a clear distinction between a solution and a non-solution. This leads to distinct ways to quantify the ability of a search method to discriminate among choices. For exact searches, one can characterize a search method by how well it selects a choice that leads closer to a solution whereas satisficing searches require discriminating among choices that lead to states of higher or lower value. For exact searches, one solution method is to prune from consideration those choices judged not to lead to a solution. In this case we can distinguish certain, so-called "admissible", search methods that are guaranteed to only prune choices that do not lead to solutions, whereas other methods can also prune paths leading to solutions, possibly causing the method to miss solutions. In the interests of clarity we will deal with just admissible, exact searches in this paper.

In the next section we describe our modelling approach and formalize the notion of a typicality criterion. We then develop analytic models of two paradigmatic search processes; depth first/find first and breadth first/find all. These models can apply to actual search algorithms ranging from naive backtracking to sophisticated heuristic methods, as discussed in Section 5. In Section 6, we test the robustness of our results by relaxing many of the assumptions under which they were derived and discover that the phenomena predicted in the simpler versions still persist in the more complicated ones too, at least within some circumscribable region of the appropriate parameter space. Finally, we place this work in context and discuss how statistical sampling techniques might be used to obtain estimates of the parameters in our model, although a quantitative evaluation of this procedure remains open.

2. Ensemble Analyses and Typicality

Search algorithms are usually intended to be run repeatedly on many different problem instances rather than on just a single case. Thus we should consider their performance over an ensemble of problem instances. Ideally, we want to calculate

the *typical* value for some property of the search for problems taken from such an ensemble, e.g. the typical cost to find the first or all solutions, whether a solution typically exists, the typical number of solutions, whether a solution can typically be found in polynomial resource bounds, etc. To do this, we need to pin down the meaning of the typicality of some property of interest. As a specific example, we focus on search cost in this paper.

Naively, one might suppose the cost can be said to have a typical value equal to its mode (most frequent) value. Unfortunately, this will not do, as there is no guarantee that the mode value is *significantly* more frequent than any other value. How about the mean (average) value? Again this will not do. Consider, for example, an ensemble consisting of instances which can mainly be solved at polynomial cost but which also possesses a few for which one would incur an exponential cost. In this case, the mean cost could be exponential and yet the cost of solving any random instance plucked from the ensemble would most likely be polynomial. Likewise the median is not necessarily typical either, as can be seen by considering a symmetric bimodal cost distribution.

The problem is plain: some distributions do not have any meaningful "typical" value. This is a sign that, for the property of interest (e.g., computational cost), not enough detail has been included in the model to specify it precisely enough. One way to quantify this is to consider the mean (which is often the most easily computed property of such models). If the relative deviation of the cost[1] is not too high, the cost distribution is quite spiked and most of the problem instances will have a cost close to the average value. To formalize this intuition, we could either pick an arbitrary threshold for the relative deviation, which would be appropriate for problems where one desires a theory for which, say, 95% of the instances are within 10% of the mean, or else we could require our average cost prediction should become more and more accurate as ensembles consisting of larger problem instances are considered. As A.I. matures the performance of our algorithms on big problems will be what matters most. So we shall use the latter option. Hence if some parameter μ, e.g. the number of variables involved in a search, describes the "scale" of our problem we can express our typicality criterion as:

[1] The relative deviation is simply the ratio of the cost's standard deviation to its mean value, i.e., $\mathrm{RelDev}(C) = \frac{\sqrt{\mathrm{var}(C)}}{\langle C \rangle}$ where $\mathrm{var}(C)$ is the variance of the cost and $\langle C \rangle$ its expected value.

> *Strong Typicality Criterion*: The average case cost $\langle C(\mu) \rangle$, which depends upon the scale parameter μ, will also be that which is typical if $\mathrm{RelDev}(C(\mu)) \to 0$ as $\mu \to \infty$.

Notice that this is a *sufficient* but not *necessary* condition; which is why we say it is "strong" (perhaps too strong). Nevertheless, it provides an effective way of checking whether an average case cost prediction will be typical.

What if the average case cost prediction for a given ensemble fails the typicality criterion? Fortunately, we are in a position to pick whatever ensemble of problems we wish to analyze. In general, changing the problem specification changes the average case complexity result. Therefore, if a given ensemble specification has an average case complexity which fails to meet our typicality criterion, we can repeatedly refine the ensemble specification until it does. Each time a refinement is made additional details about the problem instances in the ensemble are given, thereby projecting out some sub-ensemble from that originally given. This process is guaranteed to terminate because, eventually, so many details will have been specified that there is only one problem instance left that matches them all! Such extremes will generally not be encountered in practice, however, and a few iterations are usually sufficient. Each time we specify additional details the resulting model becomes more complicated. It is therefore always sensible to begin with as simple an ensemble as one can imagine and add embellishments only if absolutely necessary. This enables us to find the most parsimonious specification of a problem, sufficient to predict the typical (via mean and relative deviation) performance. We believe this idea of insisting on the typicality requirement by tightening the specification is quite novel and will help us find the simplest specification to do the job.

A subtle but important point, is that, for the same ensemble, certain questions concerning one property may have typical answers whilst those concerning another might not. For example, one could get an affirmative response to a question asking whether solutions typically exist and yet a negative response to questions concerning the typical number of solutions. Thus the question is an integral part of the typicality issue.

In summary, the analysis proceeds as follows:

1. *Specify Ensemble:* This involves identifying a small number of key parameters that describe the search problems of interest.
2. *Specify Search Process & Question:* e.g., depth first or breadth first and the question is to find the typical cost of finding one or all solutions.
3. *Calculate Average Cost:* calculate the average and variance in the number of nodes examined during the search
4. *Specify Scaling:* specify how the problem instances change as larger problems are considered
5. *Test Whether Average is Typical:* check the typicality criterion; if average is typical, stop else refine the ensemble specification and recurse.

The first step in the analysis is to construct a suitable ensemble of problem instances. Often there will be a number of plausible choices with the same basic set of parameters and the same behaviors in the limit of large problems. In such cases one is free to select that ensemble that is most easily analyzed or most readily evaluated numerically. As a concrete example, in the next section we present an ensemble for constraint satisfaction problems (CSPs). Following this we will discuss the depth first/find first and the breadth first/find all search scenarios, compute their associated costs and check typicality.

3. A Case Study: CSPs

To construct an ensemble of problem instances, begin by thinking about a single instance and then generalize to a whole family of them. A single constraint satisfaction problem [12, 11] (CSP) involves a set of μ variables, each having an associated set of domain values, together with a set of ν constraints specifying which assignments of values to variables are compatible ("good") and incompatible ("nogood"). The goal of the search is to find one or more total assignments of values to variables such that every variable has some unique value and all the constraints are simultaneously satisfied. For simplicity, in what follows we suppose each variable has the same set of domain values $\{1, \ldots, b\}$.

There are many ways to go about solving such a problem. One method is to build sequences of instantiations of variables and prune them against the constraints as and when enough variables become instantiated. This procedure naturally induces a tree structured search space, of depth μ and branching ratio b, in which a path extending from the root of the tree to the fringe corresponds to a unique solution. Partial solutions correspond to open paths awaiting further elaboration. Those partial solutions that cannot be extended to complete solutions may be pruned, thereby reducing the amount of search required. Depending on the variable and value ordering and the sophistication of the search method, partial solutions may be pruned early or late in the search.

For any CSP instance, the specific choice of nogoods completely determines which leaf nodes are solutions and which interior nodes are prunable at each level; the latter being precisely those nodes that do not lie above any solution. But now we must generalize to a whole ensemble of CSPs. To do this we can replace explicit sets of nogoods with two probabilities; p_{soln}, the probability with which each leaf node is a solution and p, the probability with which a prunable node is in fact pruned. Using these probabilities the average number of solutions for members of an ensemble of CSPs having μ variables and b values per variable is $b^{\mu}p_{soln}$ and, assuming there are say N prunable nodes at some level, the average number actually pruned at that level will be Np. In the simplest case, we will assume that p does not vary with depth in the tree and that different nodes are pruned independently. The latter assumption means that the variance in the number of nodes actually pruned at some level will be $Np(1-p)$. In more general contexts, where the pruning of different nodes is not independent we can write this variance as Npq with $q \neq 1-p$. Later, we examine the effects of relaxing the assumptions that p does not vary with depth in the tree and that the pruning is independent i.e., $q = 1-p$.

These five parameters, summarized in Table 1, allow large classes of CSPs to be characterized using a handful of degrees of freedom. A key point to emphasize is that "sophisticated" search algorithms will be those having a high p and "naive" ones, those having a lower p. Thus although we are about to describe depth first and breadth first searches, these are to be understood as no more than the basic "control" structure on top of which some (possibly clever) algorithm is running.

This is the essential trick that allows us to model a great range of algorithms at a stroke: separate the cleverness from the default control. This point is elucidated in Section 5 where we fit our "depth first/find first" model to a sophisticated graph colouring algorithm. Although the basic control flow is common throughout, the different heuristics used give rise to different effective pruning probabilities, p with correspondingly different performance characteristics. Thus our model is applicable to a very broad range of searches.

Parameter	Meaning
μ	number of variables (i.e. depth of tree)
b	number of values per variable
p_{soln}	probability with which each leaf node is a solution
p	probability with which a prunable node is in fact pruned; constant in the simplest models, but replaced by a depth dependent version later
q	characterization of pruning variance

Table 1. Parameters describing the ensemble of search problems.

Next we will examine the behaviour of two fundamental search scenarios over this ensemble. We say "scenario" rather than "algorithm" because we are especially concerned with the performance of a search algorithm *in a particular scaling limit*. Thus although we calculate the general (scale independent) cost first, the subsequent scaling specialization step is very important as different choices of scaling can affect whether or not the average cost will be typical. To illustrate this point, we will consider a depth first, find first scenario and a breadth first find all scenario in which we will pick different scaling limits.

4. Models of Naive Tree Search

Depth First/Find First

Depth-first search starts at the root and examines every state along a direct path to a leaf before backtracking and trying another path. We suppose that when

presented with a choice of branches to follow, from a particular node, each is examined in some order and the first one not pruned at that level, and not already tried, is taken. Defining z to be the effective branching ratio i.e. $z \equiv b\,(1-p)$ and assuming the pruning is uniform through the tree, the expected number of nodes that must be examined before the first solution is found or failure is established is given by:

$$\langle C \rangle = \Pr\left(N_{soln} \geq 1\right)\langle C_G \rangle + \left(1 - \Pr\left(N_{soln} \geq 1\right)\right)\langle C_N \rangle \tag{1}$$

where $\Pr\left(N_{soln} \geq 1\right) = 1 - \left(1 - p_{soln}\right)^{b^\mu}$ is the probability of having at least one solution, $\langle C_G \rangle$ is the average cost when there are solutions and $\langle C_N \rangle$ the cost when there are no solutions. In Appendix A, we show that

$$\langle C_N \rangle = \frac{b\left(1 - z^\mu\right)}{1 - z} \tag{2}$$

and

$$\langle C_G \rangle = \mu + \frac{1}{1-z}\sum_{l=0}^{\mu-1}\left(\left\langle N_l^{first}\right\rangle - 1\right)\left(1 - z^{\mu-l}\right) \tag{3}$$

where $\left\langle N_l^{first}\right\rangle$ is the average of the number of children of a node at level l that must be examined before the first one above a solution is found. This is related to the solution distribution via (see Appendix A): $\left\langle N_l^{first}\right\rangle = 1 + z\rho_l \frac{1-\rho_l^{b-1}}{1-\rho_l^b}$ where ρ_l is the probability that a node at level $l+1$ has no solutions under it, i.e., $\rho_l \equiv \left(1 - p_{soln}\right)^{b^{\mu-l-1}}$. This cost measure has a simple interpretation: At each level one must examine $\left\langle N^{first}\right\rangle - 1$ unproductive nodes, at an average cost given by $\frac{1-z^{\mu-l}}{1-z}$, before reaching the correct one (which itself adds one to the cost). Summing over all the levels of the tree gives the expression for the expected search cost.

Next we have to specify how problem instances in the ensemble should change as larger cases are considered i.e., as $\mu \to \infty$. As we are considering the case in which only a single solution is desired, we can permit the total number of solutions to grow exponentially but still remain a small fraction of the total space, i.e., we take $p_{soln} = (a/b)^\mu$ with $1 < a < b$. Using these limits the scaled cost becomes:

$$\langle C \rangle \sim \begin{cases} \mu\left(1 + \frac{z(b-1)\left(1 - \frac{\ln a}{\ln b}\right)}{(1-z)b}\right) & z < 1 \\ z^{\mu\left(1 - \ln a / \ln b\right)}\frac{K(b,z)}{z-1} & z > 1 \end{cases} \tag{4}$$

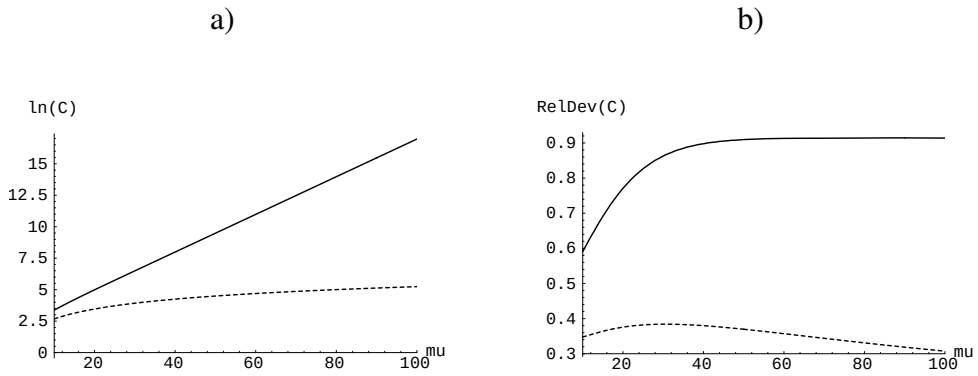

Fig. 1. Behavior of depth-first search. a) Logarithm of expected cost, $\ln(\langle C \rangle)$, versus the number of variables, μ for the case when $z = 0.8$ (dashed) and $z = 1.5$ (solid) with $b = 3$, $a = 2$. The cost grows exponentially above the transition point, $z = 1$, and polynomially below it. b) Relative deviation for these cases. It decays as μ increases for $z < 1$, so average case cost does become typical cost in this regime, but tends to a constant value for $z > 1$ suggesting that the ensemble specification needs to be refined before typical predictions can be made.

where $K(b, z)$ is a constant independent of μ (see Appendix A). Notice that the cost switches suddenly from polynomial to exponential in μ (the number of distinct variables in the CSP) as the effective branching ratio, z, increases beyond $z = 1$. This is simply understood as the point at which the pruning just balances the branching ratio from one level to the next in the tree. To understand this result qualitatively, note that the exact cost is a sum of a part that scales as μ (proportional to the number of nodes from the root directly to a solution) and another that scales as z^{μ} (which is due to pruning). Thus when z is less than one, the linear growth will dominate the cost, while above that value we get exponential growth. This gives the abruptly different scaling regimes as illustrated in Fig. 1.

Finally we test whether or not the average cost in this scaling regime is typical by calculating the behaviour of the relative deviation in the cost as $\mu \to \infty$ i.e., we check the *Strong Typicality Criterion*. We can show that the relative deviation in the cost scales in proportion to $1/\sqrt{\mu}$ for $z < 1$ and $z = b$ but is constant when $1 < z < b$. This means that, in the polynomial cost regime, the larger the problem the more typical the average cost. Whilst, in the exponential regime, additional information about the problem is required to precisely determine the typical cost, i.e., to cause the relative deviation to decay asymptotically to zero.

Without doing so, significant fluctuations about the mean cost persist even for very large problems. Thus we must refine the ensemble specification by augmenting the current description with extra information about the problem instances. As the large variance in the cost arises from fluctuations in the pruning high in the tree, a plausible contender for the missing information is the correlation in the pruning between levels in the tree. Such correlations essentially encode knowledge of the point at which the nogoods begin to have their most significant effect and therefore permit a more accurate estimation of pruning in the highest parts of the tree.

Breadth First/Find All

Next we consider a different search scenario over the same ensemble of problems; namely a breadth first search to find all solutions. Breadth first search will examine all surviving nodes at a depth l before examining any at a depth $l+1$. When any node at any level is eliminated all its descendent children are pruned too. In Appendix B we derive the average cost for this search process again under the assumption that the pruning is uniform ($p \sim$ constant). We find that:

$$\langle C \rangle = \frac{z(1-z^{\mu})}{1-z} + \frac{b-z}{1-z} \sum_{l=0}^{\mu-1} \left\langle N_l^{min} \right\rangle \left(1 - z^{\mu-l}\right) \tag{5}$$

where

$$\left\langle N_l^{min} \right\rangle = b^l \left(1 - (1-p_{soln})^{b^{\mu-l}}\right) \tag{6}$$

is the average number of unprunable nodes at each level in trees having, on average, $b^{\mu} p_{soln}$ solutions in the fringe and z is the effective branching ratio as defined above.

For this model the scaling is different. We allow the number of solutions to grow polynomially in the number of variables[2] i.e. we take $b^{\mu} p_{soln} \sim \mu^d$. With this scaling we obtain the following asymptotic form for the average cost:

$$\langle C \rangle \sim \begin{cases} \frac{(b-z)\mu^{d+1}}{1-z} & 0 < z < 1 \\ \frac{(b-z) z^{\mu} \mu^{d(1-\ln z/\ln b)} \mathcal{I}(b,z)}{z-1} & 1 < z < b \end{cases} \tag{7}$$

[2] Had we allowed it to grow exponentially, the requirement of finding them all this would always guarantee an exponential cost. Hence this is not an interesting limit.

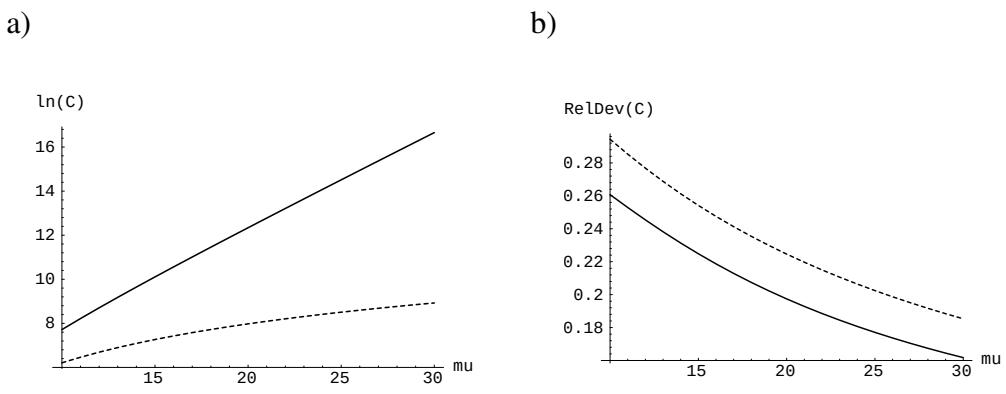

Fig. 2. Behavior of breadth-first search. a) Logarithm of expected cost, $\ln(\langle C \rangle)$, versus the number of variables, μ for the case when $z = 0.8$ (dashed) and $z = 1.5$ (solid) with $b = 3$ and $d = 1$. The cost grows exponentially above the transition point, $z = 1$, and polynomially below it. b) The relative deviation for these cases. It decays as μ increases, so average case cost does become typical cost for large problems.

where $\mathcal{I}(b, z)$ is a constant, independent of μ (see Appendix B). Therefore, we can see that for $0 < z < 1$ the cost grows polynomially with μ whilst beyond $z = 1$ the cost suddenly grows exponentially in μ leading to a phase transition phenomenon. To understand this result qualitatively, note that the exact cost contains factors proportional to the number of nodes above solutions, which will scale as μ^d, and others that scale as z^μ (due to pruning). Thus when z is less than one, the polynomial growth will dominate the cost, while above that value we get exponential growth. This gives the abruptly different scaling regimes as illustrated in Fig. 2.

On checking the *Strong Typicality Criterion*, we find that the relative deviation in the cost does go to zero as $\mu \to \infty$ for all values of $0 < z < b$ indicating that the average cost most definitely *does* become the typical cost for large problems. In this case the ensemble average can be used to estimate the cost of solving any problem instance.

5. Mapping Complex Searches to Our Model

In our models, the ability of the search heuristic to prune unproductive choices is characterized by a single parameter: the pruning probability p which we take to be uniform throughout the search. Moreover, individual pruning

choices are assumed to be independent. In Section 6 we examine the effect of relaxing these assumptions. Before doing so, however, we illustrate how this simple characterization of heuristic effectiveness applies not only to simple search methods but more sophisticated ones as well.

As a concrete example, we consider the typical behavior of a sophisticated heuristic search method, the Brelaz algorithm [18], for the graph colouring problem over the ensemble of random graphs[3]. For a given graph with μ nodes and given b colours, the problem is to find a colour for each node such that no two adjacent nodes (i.e., those at either end of an edge) have the same colour. This can be viewed as a CSP with the nodes being the variables and the colours the values to be assigned. Each edge in the graph corresponds to a binary constraint (that the linked nodes have different colours) which induces b distinct nogoods (in which nodes linked by the edge are assigned the same color). The ensemble of problems for this example consists of random graphs in which each edge exists independently with probability p_{edge}.

To make a correspondence between this problem and our model, we must specify the two parameters we used: p_{soln} which characterizes the expected number of solutions, and p which characterizes the effectiveness of the search method. Together these values determine the expected cost to find the first solution, Eq. 1. The first parameter, p_{soln}, is determined by the specification of the ensemble, i.e., from the values of $\mu, b \,\&\, p_{edge}$ as shown in Appendix C. The appropriate value for p depends also on the search method used. Once p_{soln} is known, p can be determined from the average measured[4] cost of a particular search method via Eq. 1.

For a given number of nodes, a particular choice of p_{edge} determines how many edges there will be in the graph and therefore how likely such a graph can be coloured using only b different colours. For example, if there 100 nodes in the graph and we seek a colouring using just 3 colours then setting $p_{edge} = 0.02$

[3] This ensemble contains a large number of relatively easy problems. These can be removed by considering only certain kinds of graphs [3]. Nevertheless, the ensemble of random graphs considered here illustrates how different search methods can be mapped onto our basic model.

[4] This method, which finds the precise value of p and illustrates how this single parameter can characterize a range of search methods, is computationally costly. In practice, sampling could be used to estimate its value as discussed in Section 7.

p_{edge}	**Brelaz Cost**	z
0.02	188.5	0.84
0.5	5.0	0.4

Table 2. The effective branching ratio, z, corresponding to the observed average Brelaz cost of finding the first 3–colouring of 1000 100–node random graphs for which each of their edges is present with probability p_{edge}. These values correspond to regimes in which there are very many and very few solutions, respectively.

would mean most such graphs had many possible 3–colourings. Conversely, if $p_{edge} = 0.5$ most graphs would not have any 3–colourings. To find the effective branching ratio, z, of the Brelaz algorithm for each of these regimes, we simply fit the average measured cost of the Brelaz algorithm to our generic formula by adjusting z, within the range $0 \leq z \leq 3$ (because $b = 3$) until our formula predicts a cost that agrees with the observed Brelaz cost. In the regime where there are some but not many solutions the Brelaz cost cannot be fitted to our model as the discrepancy between our theoretical prediction for the probability of solutions existing and the actual probability is quite large. This error arises because we assumed that the solutions were independent; which is not true for graph colouring. However, in regions where there are either very many or very few solutions our prediction of the probability of solutions existing is quite accurate and the fit to our model is then quite reliable. Therefore we can circumscribe a domain of applicability of our model beyond just naive depth first search in some, but not all, regions of the problem space.

6. Relaxing Modelling Assumptions

Effect of Depth Dependence in the Pruning

In the previous discussion, we assumed that the heuristic effectiveness, characterized by *p*, was the same at all depths in the search tree. However, many real searches are characterized by non-uniformities in the pruning. For example, if the pruning was based entirely on the nogoods and all the nogoods were of size 3, or larger, no pruning whatsoever could take place until at least depth 3 in the tree and moreover one would expect the pruning to increase as larger partial solutions are built up. By ignoring such irregularities we run the risk of omitting a key factor affecting both the average cost and its typicality. In this section we

therefore investigate the effects of adding depth dependence to the pruning by making $p = p_l$ a function of the level in the tree. To avoid conflating issues we consider just the case of independent pruning, $q_l = 1 - p_l$ and note that depth dependency in p induces a similar dependency in z because[5] $z_{l-1} = b(1 - p_l)$.

The actual profile of the graph colouring problem considered in Section 5 provides a realistic depth dependent profile. In Appendix C we show that the pruning probability is $p_l = 1 - \left(1 - \frac{p_{edge}}{b}\right)^{l-1}$, which rises monotonically as one descends the search tree, as shown in Fig. 3. The scaling limit considered above for the depth first search, with an exponential number of solutions on average, is obtained when the graph has relatively few edges, specifically with $p_{edge} = \frac{c}{\mu}$ with c independent of μ. This gives, asymptotically, that $z_l \sim be^{-\frac{cl}{b\mu}}$.

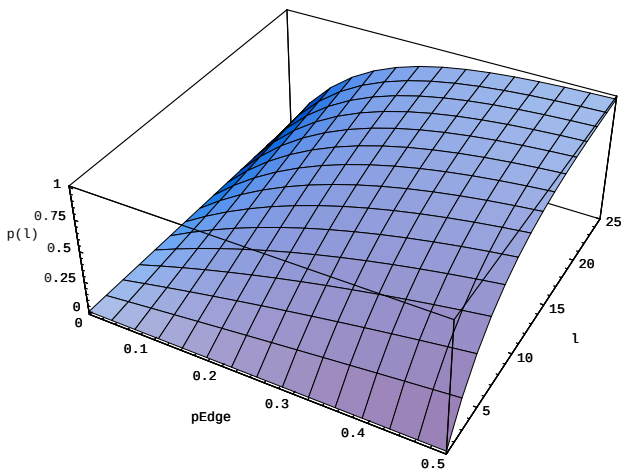

Fig. 3. Pruning probability, p_l vs. with depth, l, in the search tree, for graph colouring problems involving $\mu = 25.$ nodes, $b = 3$ colours and for which each edge exists independently with probability p_{edge}.

In terms of our models, a depth dependent pruning profile is easily incorporated in our derivation of the average costs and relative deviations by simply replacing p by p_l in the original sums. In this case, the pruning values cannot be moved outside the summations and we do not get the simple geometric sums as in the uniform case. Nevertheless, retaining the full expressions allow average costs and relative deviations to be calculated for arbitrary pruning profiles. To assess the impact of such a refinement one must test whether either the phase

[5] Here the effective branching ratio at level $l - 1$ is determined by how many children, at level l, are pruned.

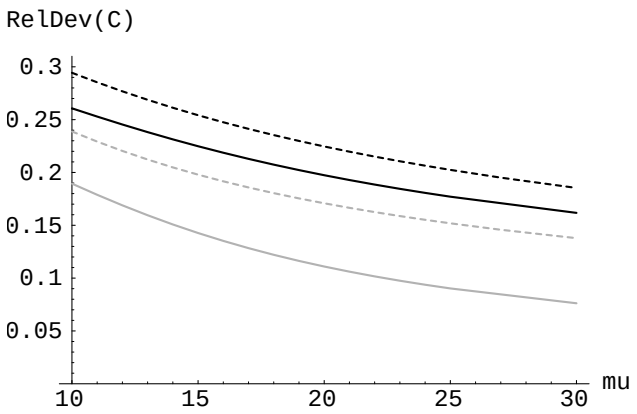

Fig. 4. Relative deviation in the cost vs. μ for a breadth first search comparing uniform pruning (dark curves, identical to those of Fig. 2b)) and depth-dependent pruning (light curves). The dashed curves correspond to $z = 0.8$ and the solid ones to $z = 1.5$. The depth-dependent profile is given by $p_l = 1 - \exp\left(-k\frac{l}{\mu}\right)$; a profile similar to that observed in graph coloring. In all cases $b = 3$ and the number of solutions scales as μ^d with $d = 1$ but for each different value of μ, k is chosen so the cost for the depth-dependent and uniform models are identical. Thus the comparison is fair. Typicality persists.

transition phenomenon or its typicality are affected. To do so, we chose to look at the breadth first search model as this has the property that the average cost is typical for all values of the effective branching ratio. The experiment consisted of picking a functional form for p_l, computing the average cost and its relative deviation using this depth dependent pruning profile, equating the cost to the cost formula for the constant pruning model and solving for the (constant) effective branching ratio that gives the same cost, finally plotting the relative deviation as predicted by the constant pruning model using this effective branching ratio. The results, shown in Fig. 4, indicate that both the phase transition phenomenon and its typicality persist (at least for all the cases we have seen). This suggests that the omission of depth dependent pruning from the simple model does not disguise any significantly new phenomenon.

Effect of Correlations in the Pruning

The preceding examples assumed that prunable nodes were pruned independently. However, this assumption will not be true in general. For example, if two different nodes in the search tree were pruned for the first time by the same nogood then their pruning must be correlated. This means that the probability

Behavior of q	**Phase Transition?**	**Is Average Typical?**	
		$z < 1$	$z > 1$
constant	yes	yes	yes
$(1-p)\ln(1+N_l)$	yes	yes	yes
$\sqrt{N_l}(1-p)$	yes	yes	no
$N_l(1-p)$	yes	yes	no

Table 3. The effect of increasing correlations in the pruning in the limit of large problems $\mu \to \infty$ whilst keeping p constant, for breadth-first search. The average cost remains typical up to moderate correlations.

of them both being pruned is higher than p^2 as would be expected if their pruning were independent. Conversely, if two nodes in the body of the tree share a common ancestor path and one of them is not pruned, the other might be a little less likely to be pruned too. We can capture the effect of a net enhancement in correlations as follows.

Let N_l be the number of prunable nodes at level l and let p be the probability with which a prunable node is pruned. In the case of independent pruning, the variance in the number of nodes actually pruned will be given by $\mathrm{Var}\left(N_l^{pruned}\right) = N_l\, p\,(1-p)$. Correlations in the pruning across a level tend to increase the variance in the number of nodes pruned. Hence, we expect $\mathrm{Var}\left(N_l^{pruned}\right) = N_l pq$ with $q \gtrsim (1-p)$. However, q cannot be arbitrarily large as can be seen by considering the most correlated possible distribution: either all of the prunable nodes are pruned (with probability p) or none of them are. This distribution has mean $N_l p$ and variance $N_l^2 p(1-p)$. This implies that $q \leq N_l\,(1-p)$. Thus, a simple model of correlation is given by making q, at a given level, depend of the number of prunable nodes at that level. Any desired functional form is possible provided this bound is not violated. The behaviors for several ranges of the variance are shown in Fig. 5 and summarized in Table 3.

Note that while the parameter q is defined in terms of the variance in the overall number of nodes pruned, it can be related to the pairwise correlation in pruning. Specifically, let $\Pr(k,n)$ be the probability that exactly k out of n nodes are pruned. This is given in terms of the more detailed distribution for a specific set s of k nodes to be selected as $\Pr(k,n) = \sum_s \Pr(s,k,n)$ where the sum is over

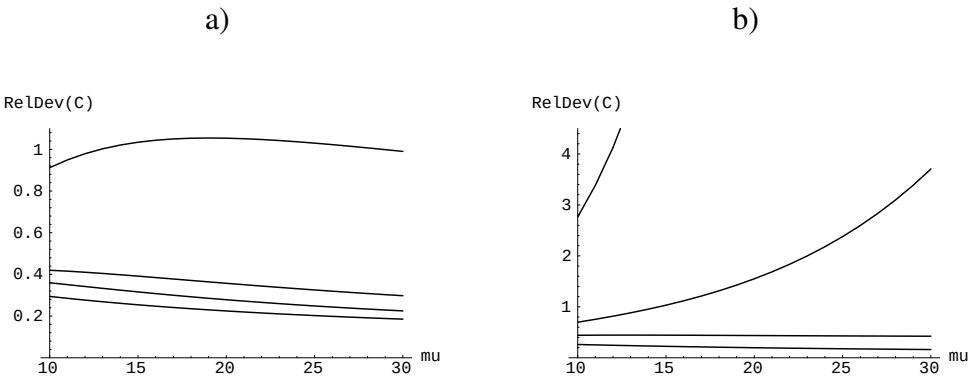

Fig. 5. The relative deviation vs. μ for the case when $b = 3, d = 1$, for breadth-first search with a) $z = 0.8$ and b) $z = 1.5$. From bottom to top, the functional forms used for q are $q = \text{constant}$, $q = (1-p)\ln(1+N_l)$, $q = \sqrt{N_l}(1-p)$ and $q = N_l(1-p)$. For each value of z, the average case costs are the same regardless of the correlations, so the phase transition phenomenon must persist. For $z < 1$ average case cost is always typical, but for $z > 1$ average cost is only typical if the correlations are not too large. In this regime a more detailed ensemble specification is required before average costs became typical.

all $\binom{n}{k}$ sets of size *k*. In terms of this distribution, we define the average and variance to be

$$\begin{aligned} np = \langle k \rangle &= \sum_k k \Pr(k,n) \\ npq = \text{var}(k) &= \sum_k k^2 \Pr(k,n) - \langle k \rangle^2 \end{aligned} \tag{8}$$

The probability that a single node *i* is pruned is given by $\Pr(i) = \sum_{i \in s} \Pr(s,k,n)$ with the sum over all sets of nodes containing *i*. Averaged over all nodes gives the (uniform) probability for a node to be pruned, i.e., $p_1 = \frac{1}{n}\sum_i \Pr(i) = \frac{1}{n}\sum_s \Pr(s,k,n)\sum_{i \in s} 1$. This last sum is just the number of elements in *s*, and so equals *k*, and the outer sum is over sets of all sizes. Thus $p_1 = \frac{1}{n}\langle k \rangle = p$.

Similarly, the joint probability that two distinct nodes *i* and *j* are pruned is given by $\Pr(i;j) = \sum_{i,j \in s} \Pr(s,k,n)$ with the sum over all sets of nodes containing both nodes. Averaged over all distinct pairs of nodes gives the (uniform)

probability for two nodes to be pruned, i.e.,

$$\begin{aligned} p_2 &= \frac{1}{n(n-1)} \sum_{i \neq j} \Pr(i;j) \\ &= \frac{1}{n(n-1)} \sum_{s} \Pr(s,k,n) \sum_{i,j \in s} 1 \end{aligned} \tag{9}$$

This last sum is just the number of distinct pairs in *s*, and so equals $k(k-1)$. The outer sum is over sets of all sizes. Thus

$$\begin{aligned} p_2 &= \frac{1}{n(n-1)} \sum_{k} k(k-1)\Pr(k) \\ &= \frac{1}{n(n-1)} \left(\langle k^2 \rangle - \langle k \rangle \right) \\ &= \frac{1}{n(n-1)} \Big(\mathrm{var}(k) + \langle k \rangle^2 - \langle k \rangle \Big) \\ &= \frac{p}{(n-1)}(q + np - 1) = p^2 + \frac{p}{(n-1)}(q - (1-p)) \end{aligned} \tag{10}$$

The (normalized) correlation in pairwise pruning is

$$\frac{p_2 - p_1^2}{p_1 - p_1^2} = \frac{1}{n-1} \, \frac{q - (1-p)}{(1-p)} \tag{11}$$

This is zero for independent choices ($q = 1 - p$) and one for the maximum variance ($q = n(1-p)$).

These findings suggest that correlations in the pruning are tolerable up to a certain level with the phase transition still located at $z = 1$. Yet, if the correlations become sufficiently large, the average cost computed can no longer be guaranteed to be typical. Again the non-zero relative deviation indicates an additional order parameter is needed to adequately characterize the search. In this case, such a parameter is the amount of correlation in the pruning across a level. We emphasize that this conclusion does not invalidate the value of ensemble analyses *per se*; it merely alerts us to the point at which a more detailed ensemble specification becomes necessary.

7. Summary and Conclusions

Standard analyses of search algorithms use detailed models of the underlying search process to determine properties of interest such as expected search cost.

While useful in specific cases, such an approach becomes increasingly difficult when applied to larger and more complex cases. There are two basic reasons for this. On the one hand, the required mathematical analysis becomes more complex and even analytically intractable, often leading to numerical simulations which can fail to convey the general nature of the qualitative phenomena. More fundamentally, a fully detailed specification of large-scale problems is often unavailable. By contrast, we have focused on determining typical behavior of a wide range of search methods based on a few parameters to characterize the search method and the set of problems.

In addition, there is the question of whether the predicted results are indeed typical. By introducing a typicality criterion, we are able to address this question and determine when further details are in fact necessary for precise predictions.

Specifically, we modeled many sophisticated searches as having a basic depth first or breadth first control structure on top of which we imposed admissible heuristics which terminate unproductive paths early or late in the search. Then we specified an ensemble of problems over which these search processes operated and estimated their typical performance via their mean performance and its relative deviation. These analyses revealed the typical presence of phase transitions in cost at a critical effective branching ratio of $z = 1$. Our model extends earlier work [8] by incorporating correlations and depth dependencies in pruning probability and by distinguishing the problem of finding one of an exponential number of solutions from that of finding all of a polynomial number; two very common CSP scenarios. Our results therefore present a more realistic picture of search problems and consequently show that the abrupt change in performance as heuristic effectiveness improves can be found in many more complex situations.

In the case of depth first/find-first search the average cost was typical when $0 < z < 1$ but not when $1 < z < b$. This result suggests that in the latter case an additional order parameter is needed. A plausible candidate is extra information about the location of the first solution in the fringe of the search tree. For the case of breadth first/find-all search, the relative deviation in the cost went to zero in all cases (assuming uncorrelated pruning) showing that average cost was always a good indicator of typical cost. However, when pruning is not independent,

sufficiently high correlations can cause the average case cost to no longer reflect typical cost. The non-zero relative deviation identifies cases in which a more refined model is necessary. Specifying additional details can reduce the variance by projecting out a sub-ensemble of more similar instances from that originally given.

How relevant are these results are to other kinds of search? The fundamental mechanism giving rise to a phase transition is competition between a branching process (exploration of a search space) and a pruning process (recognition that partial solutions should not be explored further). The competition between these two factors results in an effective branching ratio. If this effective branching ratio is greater than one then, in the limit of large searches, the cost will grow exponentially. Conversely, if the effective branching ratio is less than one, the cost will grow polynomially. This competition is so universal that similar phenomena can be expected to pervade almost any search process.

To practically exploit our formulae in the context of a given problem requires estimating the various parameters without actually doing the search on large problems. Such estimates can be obtained from a limited sampling of the search space [1, 6, 10] or from exhaustive analyses of smaller problems and using our theory to predict how the behavior scales to larger cases, e.g., examining the fluctuations which tend to get larger near phase transition points [16]. The key idea is that one can obtain information about the pruning profile by sampling random branches of the search tree and one can obtain information about the probability of solution by inspecting the ground instances of the constraints. The latter idea is developed fully in [20, 19]. Together these can be converted into an estimate of the effective branching ratio and hence the approximate search cost. Thus this theory not only provides a qualitative description of behavior of many large-scale searches, but could also be used to give quantitative estimates of the effectiveness of different search methods.

Appendix A. Depth First Search

Consider a node *n* in the tree, and let $n_1, \ldots, n_b$ denote its children nodes in the order examined by the search. The cost to process *n* with the depth-first algorithm, $C(n)$, is defined as the number of nodes, other than *n* itself, that are examined until a solution is found or it is determined that no solution exists beneath *n*. The total cost is then

$$C(n) = \sum_{i=1}^{N_{first}} (1 + C(n_i)\chi_i) \tag{12}$$

where N^{first} is the smallest *i* for which n_i is above a solution, or *b* if there are no solutions under *n*, and χ_i is 0 if n_i is a prunable node that is in fact pruned at this level, and 1 otherwise. In the sum, the 1 counts the cost of examining node n_i itself, and $C(n_i)$ is the cost to examine its children. Note that the cost for a leaf node is zero since it has no children.

It is useful to distinguish two types of nodes: those that are above at least one solution, and those that are not. Let $C_G(n)$ and $C_N(n)$ denote the costs associated with these types of nodes, respectively. For a node with no solution descendents, $N^{first} = b$ and all its children similarly have no solution descendents. Thus Eq. 12 becomes

$$C_N(n) = b + \sum_{i=1}^{b} C_N(n_i)\chi_i \tag{13}$$

On the other hand, if *n* is above a solution, so is at least one of its children giving

$$C_G(n) = N^{first} + \sum_{i=1}^{N^{first}-1} C_N(n_i)\chi_i + C_G\big(n_{N_{first}}\big) \tag{14}$$

because N^{first} is the first child above a solution.

If there are no solutions, the overall cost is then just $C_N(\text{root})$. When there are solutions, the cost is determined by the location of the first solution leaf and the pruning. The ancestor nodes of this first solution will be the "G" nodes

encountered during this search (one at each level), all nodes to the left of these in the tree will be the "N" nodes, and all other nodes will not be examined before the first solution is found. Thus the only aspect of the solution distribution relevant for the cost of this procedure is the set of values N_l^{first} for $0 \leq l < \mu$ specifying the index of the first child that is above a solution for the "G" node at level *l*. Note that the cost to examine a node with no solution descendents is determined completely by the pruning and does not depend on the solution distribution.

With independent choices for pruning, the expected cost at a node not above a solution will only depend on its level in the tree, $\langle C_N(l) \rangle$. Then the above result for nodes with no solution descendents becomes

$$\begin{aligned} \langle C_N(l) \rangle &= b + \sum_{i=1}^{b} \langle C_N(l+1) \chi_i \rangle \\ &= b + z \langle C_N(l+1) \rangle \end{aligned} \tag{15}$$

because the choice of which nodes to prune is assumed to be independent of any properties of its descendents and $\langle \chi_i \rangle = 1 - p$ (these nodes are all prunable). Since $C_N(\mu) = 0$, this immediately gives

$$\langle C_N(l) \rangle = \frac{b\big(1 - z^{\mu - l}\big)}{1 - z} \tag{16}$$

(this becomes $b(\mu - l)$ when $z = 1$).

Similarly, let $\langle C_G(l) \rangle$ denote the expected cost for the node at level *l* above the first solution. Then

$$\langle C_G(l) \rangle = N_l^{first} + \Big(N_l^{first} - 1 \Big) \langle C_N(l+1) \rangle (1 - p) + \langle C_G(l+1) \rangle \tag{17}$$

Since $C_G(\mu) = 0$, this immediately gives

$$\langle C_G(l) \rangle = \sum_{k=l}^{\mu - 1} \Big[N_k^{first} + \Big(N_k^{first} - 1 \Big) \langle C_N(k+1) \rangle (1 - p) \Big] \tag{18}$$

This is the average with respect to pruning, for a fixed location of the first solution. When averaging over the solution distribution, N_k^{first} is simply replaced by its expected value (because the cost of nodes not above a solution does not depend on the distribution of solutions).

The expected value of N_l^{first} for level l is determined by the number of children above a solution ("good" nodes) and the ability of the heuristic to select these nodes before unproductive choices. Suppose the node has n_N unproductive or "nogood" children nodes. We characterize the ability of the heuristic to order the choices by the probability[6] $1-p$ that each incorrect choice is made before a correct one. The expected number of incorrect choices made before a good node is selected is then just $n_N(1-p)$ so that $N_l^{first} = 1 + \langle n_N(l) \rangle (1-p)$.

Let ρ be the probability that each of its children has no solutions below it. Considering all possible children configurations, we have k unproductive children nodes with probability $\binom{b}{k}\rho^k(1-\rho)^{b-k}$. However, at least one child of a productive node must be above a solution so the $k=b$ case must be excluded. Thus

$$\begin{aligned} \langle n_N \rangle &= \frac{1}{1-\rho^b} \sum_{k=0}^{b-1} k \binom{b}{k} \rho^k (1-\rho)^{b-k} \\ &= b\rho \frac{1-\rho^{b-1}}{1-\rho^b} \end{aligned} \tag{19}$$

Note this has the limits

$$\langle n_N \rangle \sim \begin{cases} b\rho & \text{when } \rho \ll 1 \\ b-1-\frac{b-1}{2}(1-\rho) & \text{when } \rho \sim 1 \end{cases} \tag{20}$$

For children of nodes at level l, ρ is the probability that a node at level $l+1$ has no solutions under it, i.e., the value to use is$\rho_l \equiv (1-p_{soln})^{b^{\mu-l-1}}$.

Finally, the overall cost to find the first solution, starting from the root and assuming there is at least one solution, is just[7] $\langle C \rangle = \langle C_G(0) \rangle$, i.e.,

$$\begin{aligned} \langle C \rangle &= \sum_{l=0}^{\mu-1} \left[\left\langle N_l^{first} \right\rangle + \left(\left\langle N_l^{first} \right\rangle - 1 \right) \langle C_N(l+1) \rangle (1-p) \right] \\ &= \mu + \frac{1}{1-z} \sum_{l=0}^{\mu-1} \left(\left\langle N_l^{first} \right\rangle - 1 \right) \left(1 - z^{\mu-l} \right) \end{aligned} \tag{21}$$

[6] For simplicity, we take this probability to be the same as that characterizing whether unproductive nodes are in fact searched in regions of the tree above no solutions. If a heuristic's ability to order choices differs significantly from its ability to prune, it would be appropriate to use different quantities.

[7] Note this refers to nodes below the root, so root itself is not counted in this cost measure.

For $z = 1$ this becomes instead

$$\langle C\rangle_{z=1} = \mu + \sum_{l=0}^{\mu-1}\Big(N_l^{first} - 1\Big)(\mu - l) \tag{22}$$

1. scaling of the cost

For an exponential, but still sparse, number of solutions we take $p_{soln} = (a/b)^\mu$ with $1 < a < b$. Setting $l = L + \lambda - 1$ with $L \equiv \frac{\ln a}{\ln b}\mu$ gives $\rho_l \sim e^{-b^{-\lambda}}$.

Select $\lambda_1 < 0 < \lambda_2$ with $\ln\mu \ll |\lambda_{1,2}| \ll \mu$ (e.g., we could have $\lambda_{1,2} = \mp\sqrt{\mu}$). For $\lambda < \lambda_1$, $b^{-\lambda} \to \infty$ so ρ_l becomes very small and Eq. 20 gives $N_{first}(l) \sim 1 + ze^{-b^{-\lambda}}$. On the other hand, for $\lambda > \lambda_2$, $b^{-\lambda} \to 0$ so $\rho_l \sim 1 - b^{-\lambda}$ and $N_{first}(l) \sim 1 + z\big(1 - \frac{1}{b}\big)$.

For the scaling of the cost, we first need the sum

$$\begin{aligned}\sum_{l=0}^{\mu-1}\big(N_{first}(l) - 1\big) \sim z\sum_{l=0}^{L+\lambda_1-1} e^{-b^{L-l-1}} + \sum_{l=L+\lambda_1}^{L+\lambda_2-1}\big(N_{first}(l) - 1\big) \\ + \sum_{l=L+\lambda_2}^{\mu-1} z\left(1 - \frac{1}{b}\right)\end{aligned} \tag{23}$$

The first term is bounded by $ze^{-b^{-\lambda_1}}(L + \lambda_1) \ll \mu$ and the second by $(b-1)(\lambda_2 - \lambda_1) \ll \mu$. The third term is $z\big(1 - \frac{1}{b}\big)(\mu - L - \lambda_2) \sim z\big(1 - \frac{1}{b}\big)\big(1 - \frac{\ln a}{\ln b}\big)\mu$ which is thus the dominant contribution to the sum.

Now consider

$$\begin{aligned} s &= \sum_{l=0}^{\mu-1}\big(N_{first}(l) - 1\big)z^{\mu-l} \\ &\sim z\sum_{l=0}^{L+\lambda_1-1} e^{-b^{L-l-1}}z^{\mu-l} + \sum_{l=L+\lambda_1}^{L+\lambda_2-1}\big(N_{first}(l) - 1\big)z^{\mu-l} \\ &\quad + \sum_{l=L+\lambda_2}^{\mu-1} z\left(1 - \frac{1}{b}\right)z^{\mu-l}\end{aligned} \tag{24}$$

When $z < 1$, each term is less than the corresponding term considered above. Thus again the dominant contribution is from the third term, which equals

$z\left(1-\frac{1}{b}\right)\frac{z\left(1-z^{\mu-L-\lambda_2}\right)}{1-z} \ll \mu$, much smaller than the contribution from the previous sum. For $z > 1$, the first term is bounded by $b^{\mu+1}e^{-b^{-\lambda_1}}(L+\lambda_1) \to 0$ because $|\lambda_1| \gg \ln\mu$. The third term is now asymptotically $z\left(1-\frac{1}{b}\right)\frac{z^{\mu-L-\lambda_2+1}}{z-1} \ll z^{\mu-L+1}$. The second term can be written as $z^{\mu-L+1}\sum_{\lambda=\lambda_1+1}^{\lambda_2}(K_\lambda-1)z^{-\lambda}$ where K_λ is the value of $N_{first}(l)$ obtained using $\rho_l \sim e^{-b^{-\lambda}}$, showing that this term dominates the sum. Thus in this case $s \sim z^{\mu(1-\ln a/\ln b)}K(b,z)$ with $K(b,z) \equiv z\int_{-\infty}^{\infty}d\lambda\, z^{-\lambda}(K_\lambda-1)$.

Using these limits in Eq. 21 gives the final scaled average cost of Eq. 4.

Appendix B. Breadth First Search

In breadth first search, all nodes at a given level in the tree are examined, with those that cannot be pruned then expanded to the next level. Consider the breadth-first search at level *l*. Let N_l^{after} be the number of nodes at this level after removing any found to be nogood. Since each surviving node at the previous level generates *b* nodes at this level, the search will need to examine bN_{l-1}^{after} nodes at level l giving a total cost of

$$C = b\sum_{l=0}^{\mu-1} N_l^{after} \tag{25}$$

The expected cost is then determined by the expected number of nodes after pruning at each level. The number of nodes surviving at a given level after pruning must include all unprunable nodes plus those of the prunable ones that were not eliminated. Let $P_k^{(l)}$ be the probability that there are *k* nodes at level l after pruning, i.e., $N_l^{after} = k$. The pruning can be characterized by $P_{prune}(N, r)$, the probability that *r* nodes are in fact pruned in a level with *N* prunable nodes, whose mean is given by Np thus defining the uniform pruning probability *p*. Thus

$$\left\langle N_l^{after} \right\rangle = \sum_k kP_k^{(l)} = \sum_m P_m^{(l-1)} \sum_k kP_{km}^{(l)} \tag{26}$$

where $P_{km}^{(l)} = P_{prune}\big(bm - N_l^{min}, bm - k\big)$ is the probability that $N_l^{after} = k$ *given* that $N_{l-1}^{after} = m$. With the definition of *p*, we have

$$\begin{aligned}
\sum_k kP_{km}^{(l)} &= bm\sum_k P_{km}^{(l)} - \sum_{k=N_{min}(l)}^{bm} (bm-k)P_{km}^{(l)} \\
&= bm - \sum_{k=0}^{bm-N_{min}(l)} kP_{prune}\big(bm - N_l^{min}, k\big) \\
&= bm - \big(bm - N_l^{min}\big)p \\
&= bm(1-p) + pN_l^{min}
\end{aligned} \tag{27}$$

Thus we obtain the recursion relation

$$\begin{aligned}
\left\langle N_l^{after} \right\rangle &= b(1-p)\sum_m mP_m^{(l-1)} + pN_l^{min} \\
&= b(1-p)\left\langle N_{l-1}^{after} \right\rangle + pN_l^{min}
\end{aligned} \tag{28}$$

This model allows for the possibility of having no solutions so the root itself could be pruned. In particular, if $N_0^{min} = 1$ (i.e., there is at least one solution), the root cannot be pruned so $N_0^{after} = 1$. On the other hand, if $N_0^{min} = 0$ (no solutions), the root will be pruned with probability p in which case N_0^{after} will be zero, otherwise one. Hence $\left\langle N_0^{after} \right\rangle = 1 - p + pN_0^{min}$ and the recursion relation gives

$$\left\langle N_l^{after} \right\rangle = z^l(1-p) + p\sum_{k=0}^{l} z^{l-k} N_k^{min} \tag{29}$$

where $z \equiv b(1-p)$.

From this we obtain the average cost for a given solution set:

$$\begin{aligned} \frac{\langle C \rangle}{b} &= \sum_{l=0}^{\mu-1} z^l(1-p) + p\sum_{l=0}^{\mu-1}\sum_{k=0}^{l} z^{l-k} N_k^{min} \\ &= \frac{z(1-z^\mu)}{b(1-z)} + \frac{b-z}{b(1-z)}\sum_{k=0}^{\mu-1} N_k^{min}\left(1 - z^{\mu-k}\right) \end{aligned} \tag{30}$$

The only quantity in this result that depends on the solution set is the size of the minimum sets. Thus a final average over the solution sets gives

$$\langle C \rangle = \frac{z(1-z^\mu)}{1-z} + \frac{b-z}{1-z}\sum_{k=0}^{\mu-1} \left\langle N_k^{min} \right\rangle \left(1 - z^{\mu-k}\right) \tag{31}$$

Note that this derivation is valid for any pruning and solution distributions, i.e., only their averages enter into the determination of the average cost. This fact is used to examine the effect of correlations.

1. scaling of the cost

For a polynomial number of solutions, $\langle S \rangle \sim \mu^d$, we have $p_{soln} \sim \mu^d / b^\mu$ so

$$\ln\left(1 - p_{soln}\right)^{b^{\mu-l}} \sim b^{\mu-l}\left(-\frac{\mu^d}{b^\mu} + \ldots\right) = -\frac{\mu^d}{b^l} + \ldots \tag{32}$$

Thus $\left(1 - p_{soln}\right)^{b^{\mu-l}} \sim e^{-\mu^d/b^l}$ with the correction being exponentially small. Finally we get from Eq. 6 that $\langle N_{min}(l) \rangle \sim b^l\left(1 - e^{-\mu^d/b^l}\right)$. Let $L \equiv \frac{d}{\ln b}\ln \mu$.

Then μ^d/b^l goes to zero or infinity as $\mu \to \infty$ depending on whether *l* is greater or less than *L*, respectively. Thus for small *l*, $\langle N_{min} \rangle \sim b^l$ and for large *l*, $\langle N_{min} \rangle \sim \mu^d$, a constant independent of *l*. Thus the expected number of nodes above at least one solution initially grows exponentially but eventually reaches a constant value given by the total number of solutions.

We need the asymptotic value of the sum $s \equiv \sum_{l=0}^{\mu-1} b^l \Big(1 - e^{-\mu^d/b^l}\Big)\big(1 - z^{\mu-l}\big)$. The asymptotic behavior of this sum depends on the size of *z*. When $z < 1$, select $L \ll \lambda \ll \mu$, so that

$$s \sim \sum_{l=0}^{\lambda} \langle N_{min}(l) \rangle \Big(1 - z^{\mu-l}\Big) + \sum_{l=\lambda+1}^{\mu-1} \mu^d - \sum_{l=\lambda+1}^{\mu-1} \mu^d z^{\mu-l} \tag{33}$$

The first term is bounded by $\mu^d \lambda \ll \mu^{d+1}$. The second term is equal to $\mu^d(\mu - \lambda - 1) \sim \mu^{d+1}$ and the third term is $-\mu^d z^{\mu-\lambda-1} \sum_{l=0}^{\mu-\lambda-2} z^{-l} \sim -\frac{z\mu^d}{1-z} \ll \mu^{d+1}$. Thus we have $s \sim \mu^{d+1}$ in this case. (Note this assumes *z* is fixed, but the limit is not uniform, i.e., when *z* is closer to one, larger μ will be required before the error terms become small.)

When $1 < z < b$, it is useful to decompose the sum $s = \sum \langle N_{min}(l) \rangle - s_1$ where $s_1 \equiv \sum \langle N_{min}(l) \rangle z^{\mu-l}$. As shown for the $z < 1$ case, the first term scales as μ^{d+1}. In s_1 for μ fixed, the terms grow exponentially with *l* as $z^\mu (b/z)^l$ for small *l*, and decrease exponentially as $\mu^d z^{\mu-l}$ for large *l*, hence reaching a maximum near *L*. Thus we expect the dominant contribution to the sum from *l* values of the form $l = L + \lambda$ where λ ranges over both positive and negative values, but is independent of μ. In this regime, $\mu^d/b^l \sim b^{-\lambda}$. (Note this continuously interpolates between the two limits described above for $l \ll L$ and $L \ll l$.) In this range of *l* values, the terms in s_1 become $z^\mu \big(\frac{b}{z}\big)^l \Big(1 - e^{-b^{-\lambda}}\Big)$ so

$$\begin{aligned} s_1 &\sim z^\mu \left(\frac{b}{z}\right)^L \sum_{\lambda} \left(\frac{b}{z}\right)^\lambda \Big(1 - e^{-b^{-\lambda}}\Big) \\ &\sim z^\mu \mu^{d(1 - \ln z/\ln b)} I(b,z) \end{aligned} \tag{34}$$

where $I(b,z) \equiv \int_{-\infty}^{\infty} d\lambda \; (b/z)^\lambda \Big(1 - e^{-b^{-\lambda}}\Big)$ is a constant independent of μ (extending the range of summation to infinity introduces exponentially small

corrections). Because this term grows exponentially with μ, we finally have $s \sim -z^\mu \mu^{d(1-\ln z/\ln b)} I(b,z)$.

When $z = b$, this derivation no longer holds: the terms of the sum are strictly decreasing, so the smallest l values dominate the sum. Specifically for small l, the terms in s_1 are almost constant, equal to b^μ, while for large l, they decay exponentially as $\mu^d b^{\mu-l}$. Select $\lambda_1,\ \lambda_2$ so that $L - \lambda_1 \gg 1$, $\lambda_2 - L \gg 1$ but with $\lambda_2 - \lambda_1 \ll L$ (e.g. we could have $\lambda_{1,2} = L \mp \sqrt{\ln \mu}$) to get

$$s_1 \sim \sum_{l=0}^{\lambda_1} b^\mu + \sum_{l=\lambda_1+1}^{\lambda_2} \langle N_{min}(l)\rangle b^{\mu-l} + \sum_{l=\lambda_2+1}^{\mu-1} \mu^d b^{\mu-l} \tag{35}$$

The first term is $b^\mu(\lambda_1+1) \sim b^\mu L$, the second is at most $b^\mu(\lambda_2-\lambda_1) \ll b^\mu L$. The third term is $\mu^d b^{\mu-\lambda_2-1} \sum_{l=0}^{\mu-\lambda_2-2} b^{-l} = b\mu^d \frac{b^{\mu-\lambda_2-1}-1}{b-1}$ which scales as $\frac{b^{\mu-\lambda_2}\mu^d}{b-1} \ll \frac{b^{\mu-L}\mu^d}{b-1}$. Now $\mu^d/b^L = 1$ so this term is in fact much less than b^μ. Thus we have $s_1 \sim b^\mu L$ and hence $s \sim -\frac{d}{\ln b} b^\mu \ln \mu$ which is asymptotically much larger than the $z < b$ case.

Finally there is the special case $z = 1$ in which case $s = 0$ but we really need $s/(1-z)$ for the cost. This quantity becomes $s' = \sum \langle N_{min}(l)\rangle(\mu - l)$. If we select $L \ll \lambda \ll \mu$, then

$$s' \sim \sum_{l=0}^{\lambda} \langle N_{min}(l)\rangle(\mu-l) + \sum_{l=\lambda+1}^{\mu-1} \mu^{d+1} - \sum_{l=\lambda+1}^{\mu-1} \mu^d l \tag{36}$$

The first term is bounded by $(\lambda+1)\mu^{d+1} \ll \mu^{d+2}$. The second term is equal to $\mu^{d+1}(\mu-1-\lambda) \sim \mu^{d+2}$ and the third is $-\frac{\mu^d}{2}(\mu(\mu-1) - \lambda(\lambda-1)) \sim -\frac{\mu^{d+2}}{2}$. Thus $s' \sim \mu^{d+2}/2$.

Using these limits in Eq. 5 gives the final scaled average cost (using $bp = b - z$):

$$\langle C\rangle \sim \begin{cases} \frac{(b-z)\mu^{d+1}}{1-z} & z < 1 \\ (b-1)\mu^{d+2}/2 & z = 1 \\ \frac{(b-z)z^\mu \mu^{d(1-\ln z/\ln b)} I(b,z)}{z-1} & 1 < z < b \\ \frac{b^{\mu+1}}{b-1} & z = b \end{cases} \tag{37}$$

(recall that $z = b$ corresponds to $p = 0$).

Appendix C. Example: Graph Coloring

In this appendix, we derive the pruning profile and expected number of solutions for the graph coloring example which requires assigning one of b values to each of μ variables (or nodes in a graph), such that no nodes linked by an edge have the same value. We consider the simple ensemble of problems obtained by using random graphs in which each edge exists with probability $p_{edge} = \frac{c}{\mu}$, for some constant c, so the expected number of edges is $\frac{c(\mu-1)}{2}$. Note that an edge linking variables i and j gives the b nogoods $\{v_i = \alpha,\ v_j = \alpha\}$ for $\alpha = 1, \ldots, b$ with v_i being the value assigned to variable i.

Consider a fixed ordering for the variables and level l in the corresponding search tree. We need the probability that a node at this level is pruned given that the search process arrives at its parent node (i.e., no ancestors were pruned by the choice of edges in the graph). Pruning in such a case requires a nogood that matches the assignment of variable l and that of one of the previous assignments. Thus the probability the node is not pruned is just $\left(1 - p_{edge}\right)^{n_{v_l}}$ where n_{v_l} is the number of assignments among the levels before l that have the same assigned value as that at level l (if an edge exists between any such variables, a nogood will prune it). This must be averaged over the nodes at this level to obtain an average pruning value for the level:

$$\begin{aligned} 1 - \bar{p} &= \frac{1}{b^l} \sum_{v_1 \ldots v_l} \left(1 - p_{edge}\right)^{n_{v_l}} \\ &= \frac{1}{b^l} \sum_{v_l=1}^{b} \sum_{n_1 \ldots n_b} \left(1 - p_{edge}\right)^{n_{v_l}} \sum_{v_1 \ldots v_{l-1}} 1 \end{aligned} \tag{38}$$

with the last sum over all assignments to prior levels that satisfy the condition of having n_i assignments of value i. This last sum is just the number of ways to put $l-1$ items into b boxes such that box i has n_i, i.e., a multinomial coefficient. Thus we have

$$\begin{aligned} 1 - \bar{p} &= \frac{1}{b^l} \sum_{v_l=1}^{b} \sum_{n_1 \ldots n_b} \binom{l-1}{n_1 \ldots n_b} \left(1 - p_{edge}\right)^{n_{v_l}} \\ &= \frac{1}{b^l} \sum_{v_l=1}^{b} \left(b - p_{edge}\right)^{l-1} = \left(1 - \frac{p_{edge}}{b}\right)^{l-1} \end{aligned} \tag{39}$$

Asymptotically as $\mu \to \infty$ this becomes $z_{l-1} = b(1-\bar{p}) \sim be^{-\frac{c}{b}\frac{(l-1)}{\mu}}$ since this argument applies to pruning at level *l*, and hence branching ratio at level $l-1$.

For the expected number of solutions, we must consider the probability that a leaf node, or any ancestor, is pruned by a nogood. Since nogoods eliminate identical assignments for variables linked by an edge, the probability that a leaf is a solution is just[8] $\left(1-p_{edge}\right)^{\sum_{i<j}\delta_{v_i v_j}}$ since the exponent is the number of pairs of variables (each counted only once since edges are undirected) with the same assignments: each of these must not have an edge between them in order for the leaf to be unpruned, i.e., be a solution. Summing over all leaves, i.e., all possible value assignments, gives the expected number of solutions for the problem.

To get an asymptotic expression we have $\sum_{i<j}\delta_{v_i v_j} = \frac{1}{2}\left(\sum_{\alpha} n_\alpha^2 - \mu\right)$ where n_α is the number of variables having assignment α. Let $x \equiv \left(1-p_{edge}\right)^{1/2} \sim e^{-c/2\mu}$. Then the average number of solutions is given by

$$s = x^{-\mu} \sum_{n_1 \ldots n_b} \binom{\mu}{n_1 \ldots n_b} x^{n_1^2+\ldots+n_b^2} \tag{40}$$

This is approximated by a gaussian integral by taking logs of the terms in the sum and expanding around the maximum $n_\alpha = \frac{\mu}{b} + \delta_\alpha$ with $\sum \delta_\alpha = 0$. This gives an asymptotic expression for the logs of the terms as $A - \frac{b+c}{2\mu}\sum \delta_\alpha^2$ with

$$A = \mu\left(\ln b - \frac{c}{2b}\right) - \frac{b-1}{2}\ln\mu + \frac{b\ln b}{2} - \frac{b-1}{2}\ln\left(2\pi\right) \tag{41}$$

Hence we have

$$s \sim e^{A+c/2} \int_{\{\delta_\alpha\}} d\delta_\alpha \; e^{-\frac{b+c}{2\mu}\sum \delta_\alpha^2} \tag{42}$$

Evaluation of the integral gives

$$\left(\frac{2\pi\mu}{b+c}\right)^{\frac{b-1}{2}} \frac{1}{\sqrt{b}} \tag{43}$$

[8] Here we use δ_{xy} which is one when $x = y$ and zero otherwise.

so

$$\begin{aligned} s &\sim e^{\mu\left(\ln b - \frac{c}{2b}\right)} e^{\frac{b-1}{2}\ln(b/2\pi)} \left(\frac{2\pi}{b+c}\right)^{\frac{b-1}{2}} \\ &\sim e^{\mu\left(\ln b - \frac{c}{2b}\right)} \left(\frac{b}{b+c}\right)^{\frac{b-1}{2}} \end{aligned} \tag{44}$$

Note that when $c = 0$ (no edges in graph so all states are solutions) this gives b^{μ}. Also, it predicts a change from exponentially many to exponentially few solutions (on average) at the value $c_{crit} = 2b \ln b$.

References

[1] Bruce Abramson. *The Expected-Outcome Model of Two-Player Games*. PhD thesis, Columbia University, 1987.

[2] M. Bruynooghe. Solving combinatorial search problems by intelligent backtracking. *Information Processing Letters*, 12:36–39, 1981.

[3] 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.

[4] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, NY, 1989.

[5] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Clarendon Press, Oxford, 2nd edition, 1992.

[6] J. M. Hammersley and K. W. Morton. Poor man's monte carlo. *J. Roy. Statist. Soc. Ser. B*, 16:23–28, 1954.

[7] Theodore E. Harris. *The Theory of Branching Processes*. Dover, New York, 1989.

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

[9] R. M. Karp and J. Pearl. Searching for an optimal path in a tree with random costs. *Artificial Intelligence*, 21:99–116, 1983.

[10] Donald E. Knuth. Estimating the efficiency of backtrack programs. *Mathematics of Computation*, 29:121–136, January 1975.

[11] Vipin Kumar. Algorithms for constraint-satisfaction problems: A survey. *AI Magazine*, 13(1):32–44, 1992.

[12] A. K. Mackworth. Constraint satisfaction. In S. Shapiro and D. Eckroth, editors, *Encyclopedia of A.I.*, pages 205–211. John Wiley and Sons, 1987.

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

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

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

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

[17] H. S. Stone and P. Sipala. The average complexity of depth-first search with backtracking and cutoff. *IBM Journal of Research and Development*, 30:242–258, 1986.

[18] Jonathan S. Turner. Almost all k-colorable graphs are easy to color. *Journal of Algorithms*, 9:63–82, 1988.

[19] 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.

[20] Colin P. Williams and Tad Hogg. Exploiting the deep structure of constraint problems. *Artificial Intelligence*, 70:73–117, 1994.