Statistical Mechanics of Combinatorial Search

Tad Hogg
Xerox Palo Alto Research Center
Palo Alto, CA 94304
hogg@parc.xerox.com

Abstract

The statistical mechanics of combinatorial search problems is described using the example of the well-known NP-complete graph coloring problem. A simple parameter describing the problem structure predicts the difficulty of solving the problem, on average. However, because of the large variance associated with this prediction, it is of limited direct use for individual instances. Additional parameters, describing problem structure as well as heuristic effectiveness, are introduced to address this issue. This also highlights the distinction between the statistical mechanics of combinatorial search problems, with their exponentially large search spaces, and physical systems, whose interactions are often governed by a simple euclidean metric.

1: Introduction

Combinatorial search is among the hardest of common computational problems: the solution time can grow exponentially with the size of the problem [8]. Examples arise in scheduling, planning, circuit layout, spin glasses and machine vision, to name a few areas. Fundamentally, the problem consists of finding those combinations of a discrete set of items that satisfy specified requirements, e.g., minimizing a cost function. The number of possible combinations to consider grows very rapidly (e.g., exponentially or factorially) with the number of items, leading to potentially lengthy solution times and severely limiting the feasible size of such problems.

In practice, heuristics [24] are often used to select the next combination, or state, to consider. Typically, a heuristic evaluates a small number of potential changes to the current state, based on easily computed local properties of the state, e.g., by considering only some of the requirements. Often this leads to a series of choices that produces a solution much faster than uninformed random selection. Unfortunately any such evaluation, based on local information, can also be misleading with respect to the overall, or global, requirements.

When faced with such a search problem, we would like to know which heuristic is likely to be best, or determine whether the problem is solvable with available methods and current hardware speeds. This goal would be difficult to achieve if each problem or search method were very different from others. Fortunately, however, recent studies [2, 5, 9, 17, 18, 22, 25, 29, 30] have made considerable progress in this direction. Specifically, for large problems, a few parameters characterizing the structure of the search problem determine the difficulty for a wide variety of common heuristics, on average. Moreover, there are transitions, becoming more abrupt for larger problems, as these parameters vary. These are analogous to phase transitions in physical systems and identify situations in which major gains are possible from small improvements in the local heuristic evaluations.

Here, these results are summarized for one type of combinatorial search, constraint satisfaction [20], using the particular problem of graph coloring. While this work is encouraging, its use is limited by the large variances that remain even after the structural parameters are specified. As a step toward improving this situation, additional parameters, for problem structure and heuristic effectiveness, are presented. Finally, some applications and open issues are discussed.

2: Graph coloring

A graph coloring problem consists of a graph, a specified number of colors, and the requirement to find a color for each node in the graph such that adjacent nodes (i.e., nodes linked by an edge in the graph) have distinct colors. Many important A.I. problems, such as planning and scheduling, can be mapped onto the graph coloring problem. Moreover, as a well-known NP-complete problem, graph coloring has received considerable attention [21, 14, 26]. For simplicity, we consider the ensemble of problems given by random graphs [1], i.e., taking each graph with the specified number of nodes and edges to be equally likely.

The experiments presented below used a complete, depth-first backtracking search based on the Brelaz heuristic [14] which assigns the most constrained nodes first (i.e., those with the most distinctly colored neighbors), breaking ties by choosing nodes with the most uncolored neighbors (with any remaining ties broken randomly). For each node, the smallest color consistent with the previous assignments is chosen first, with successive choices made when the search is forced to backtrack. We measure the search cost by the number of states in the search tree that are expanded until the first solution is found or, when there are no solutions, until no further possibilities remain to be examined. As a simple optimization, we never change the colorings for the first two nodes selected by this heuristic. Any such changes, which could only occur when the backtrack search has failed to find a solution starting from the initial assignments for the first two nodes, would amount to unnecessarily repeating the search with a permutation of the colors.

3: Problem structure and search cost

For graph coloring, the average degree of the graph γ (i.e., the average number of edges coming from a node in the graph) is an structural parameter that distinguishes relatively easy from harder problems, on average. This parameter, also called the connectivity, is related to the number of edges e and number of nodes n in the graph by $e = \frac{1}{2}\gamma n$. In this paper, we focus on the case of 3–coloring (i.e., when 3 different colors are available).

The relation between the graph's structure and the number of search steps required to color it, or determine no coloring is possible, is shown in Fig. 1. Specifically, this shows that sparse and dense graphs are typically easy to color while those with an intermediate number of edges are more difficult. The observed peak for random graphs is at $\gamma = 4.6$. Note this is slightly lower than the value of 5.1 for graphs not reducible with respect to a variety of local simplifications [2]. As shown in the figure, this peak also coincides with the point at which the fraction of graphs with a solution drops from near one to near zero. For graphs with more nodes, the peak in search cost becomes sharper and the fraction of soluble cases drops more abruptly.

This observation associates a region with a high density of relatively hard problems with an abrupt transition in the nature of the problems themselves. That is, the drop in the fraction of soluble cases represents a transition from underconstrained graphs with many solutions to overconstrained ones with none. In the transition region, graphs typically have many large partial solutions

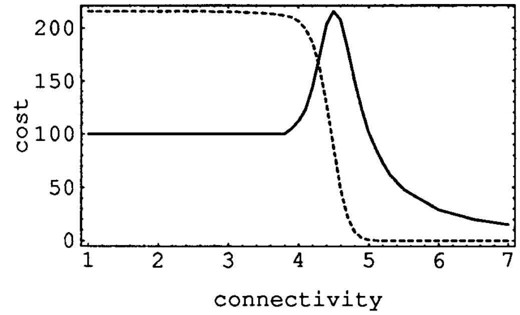

Fig. 1. Behavior for 3–coloring of random graphs with 100 nodes as a function of connectivity γ in steps of 0.1. The solid curve shows the median search cost, and the dashed one is the fraction of graphs with a solution (ranging from one on the left to zero on the right).

(i.e., consistent ways to color large subsets of the graph) but few, if any, complete solutions.

This behavior can be quantified with the following approximate argument [2, 30, 27]. Let b be the number of available colors. A given edge in the graph eliminates b of the b^2 possible ways to color the nodes it connects. So a choice of colors will satisfy the constraint with probability $1-\frac{1}{b}$. Consider a state in which k nodes of the graph are colored. There are b^k possible colorings for this subgraph, and the probability a given edge will be within this subgraph is $\binom{k}{2}/\binom{n}{2} \sim \left(\frac{k}{n}\right)^2$. Thus a coloring of the k nodes will violate the constraint of a given edge with probability $\left(\frac{k}{n}\right)^2\frac{1}{b}$. Assuming independence among the coloring constraints of the different edges, there will be, on average,

$$N_k = b^k\left(1 - \left(\frac{k}{n}\right)^2\frac{1}{b}\right)^e \tag{1}$$

consistent colorings for these nodes.

The behavior of N_k, shown in Fig. 2, qualitatively explains the search behavior of Fig. 1. Specifically, each step of a backtrack-based search method, as used here, attempts to extend a partially colored subgraph to consistently include one more node, backtracking when there are no available colors for the new node. From the figure, we see that when there are few edges, N_k increases monotonically, so on average there will be at least one consistent way to color the next node considered. This means that a typical backtrack search will almost always be able to extend the partially colored subgraph, and go directly to a solution with little or no backtracking. As more edges are added, the number of solutions, N_n, decreases very rapidly, while the number of partial colorings for smaller parts of the graph decreases less rapidly. This leads to a maximum in N_k at a value $k_{max} < n$. In this situation, a typical search will proceed with little backtrack up to $k \approx k_{max}$ but is then unlikely to be able to proceed further. Thus we can expect a great deal of backtracking until the search finds one of the relatively rare partial colorings that does lead to a solution. This leads to an increase in search cost, which continues as long as the number of solutions drops more rapidly than the number of partial solutions at k_{max}. Finally, when there are very many edges, there are unlikely to be any solutions in which case all partial solutions must be examined in the search. However, as seen in Fig. 2, the number of partial solutions, $\sum_k N_k$ decreases as edges are added resulting in a lowered overall search cost.

This qualitative description of the behavior of the search cost predicts that the hardest problems will occur at about the same point as when the number of solutions finally drops to zero, explaining the correspondence between the search cost peak and the drop in the probability for a solution to exist. Another observation is that when there is little or no backtracking (i.e., when N_k is growing monotonically), the search cost should grow only linearly with the size of the problem, n. Otherwise, we can expect exponential growth in the search cost as n increases, due to the exponentially increasing difference between the number of partial colorings, especially $N_{k_{max}}$, and the number of solutions.

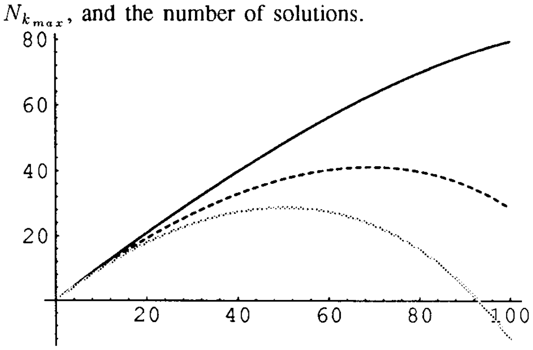

Fig. 2. Number of partial colorings as a function of their size. Specifically, this shows $\ln N_k$ vs. k for $n = 100$ and $b = 3$ for $\gamma = 1.5$ (solid), 4 (dashed) and 6 (gray).

We can also use Eq. 1 to estimate the regions with these different behaviors. First, N_k increases monotonically as long as $\gamma < (b-1)\ln b$, or 2.2 for $b = 3$. Second, a rough criterion for when the number of solutions just reaches zero is $N_n = 1$, i.e., at $\gamma = \frac{-2\ln b}{\ln(1-1/b)}$, or 5.4 for $b = 3$. For larger or smaller γ, N_n approaches zero or infinity, respectively, as n increases.

In addition, for sufficiently large γ, the peak in the number of partial states will be at $k_{max} \ll n$ which would give a regime of polynomial growth in the search cost. In this regime

$$\ln N_k \sim k \ln b - e\left(\frac{k}{n}\right)^2 \frac{1}{b} \tag{2}$$

with maximum at $k_{max} = \frac{n^2}{2e} b \ln b$ or $k_{max} = \frac{n}{\gamma} b \ln b$, and $\ln N_{k_{max}} \sim \frac{n}{2\gamma} b(\ln b)^2$. Thus we can expect polynomial search cost when this grows logarithmically, i.e., when $\gamma = O(\frac{n}{\ln n})$ (corresponding to edge probability of $\frac{\gamma}{n} = O(\frac{1}{\ln n})$) This suggests that dense graphs are easy to color, or determine that no coloring exists, which is also the case when restricting attention to graphs that do have a coloring [28, 7].

This theory ignores edge dependencies and the ability of heuristics to color nodes in an order that increases the likelihood of early pruning. However, it describes the essential mechanism driving this transition from under- to overconstrained problems and explains why hard cases are likely to be found in the transition region. In more quantitative terms, it predicts an observed transition from linear to exponentially growing search cost [12] and determines roughly the parameter values at which these transitions take place. The peak in the search cost also appears [30] for methods, such as heuristic repair [21] or simulated annealing [16], that incrementally modify a completely colored graph in an attempt to remove conflicts. For these methods the peak in search cost is due to changes in the relative density of solutions among the complete colorings whose number of conflicts cannot be reduced by a single change.

4: The search cost distribution

The previous section described an easy-hard-easy transition in search cost and gave a simple theoretical explanation for this behavior. However, a more complex story is seen from the full distribution of search costs. Surprisingly, as shown in Fig. 3, exceptionally hard instances are concentrated not around the peak in the median, but rather at lower connectivities [12]. Thus, for 100–node graphs, $\gamma = 4.5$, near the median peak, gives many more cases with cost above 1000 than $\gamma = 3$, but the reverse is true for costs above 100,000. This suggests that there are actually two qualitatively distinct regions of hard problems: 1) a region containing a high density of relatively hard problems giving the peak in the median, and 2) a region, with somewhat lower connectivity, in which most

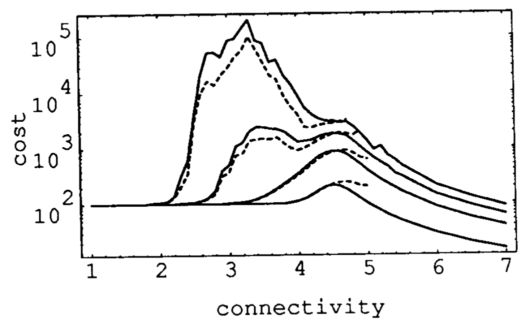

Fig. 3. Cost percentiles vs. connectivity for 100–node graphs, based on 50000 samples at each value of γ, given in increments of 0.1. Two sets of curves are shown: solid, for the behavior of all samples, and dashed, for those samples with solutions. The dashed curves extend only up to $\gamma = 5$ since beyond that point few instances have solutions. For each set of curves, the lowest shows the 50% cost (i.e., the median). Successively higher curves show the costs for the top 0.05, 0.005 and 0.0005 of the problems.

problems are very easy but which also contains a few exceptionally hard instances.

This behavior also gives rise to a huge variation in search cost, which is observed to persist as the size of the problem increases. This is unlike the usual cases in statistical physics where the relative fluctuations in thermodynamic observables go to zero as $\frac{1}{\sqrt{n}}$. Thus in order to have confidence that most observed searches will be fairly close to the expected theoretical behavior, a more detailed description of the search problems is required. This description can consist of additional information about the problem structure (e.g., to more precisely specify the number of partial colorings of different sizes) as well as a characterization of the effectiveness of the heuristic search method in avoiding unproductive choices. These possibilities are discussed in the next two sections.

5: Characterizing problem structure

While Eq. 1 gives the expected number of partial colorings, there is a large variation among graphs with a given number of edges. One way to make more precise predictions of these values and hence obtain a more sensitive measure of problem difficulty, is to use additional information about the graph.

We can proceed in a systematic manner to obtain additional structural parameters. Consider a given coloring problem with b colors, n nodes and e edges. Let E_k be the set of colorings for the graph that are eliminated by edge k (namely, all those colorings for which the nodes connected by that edge have the same color). We define the size of the intersections of these sets

$$S_r = \sum |E_{l_1} \cap \ldots \cap E_{l_r}| \tag{3}$$

where the sum is over all r-subsets of $\{1, \ldots, e\}$, and we define $S_0 = b^n$, the total number of possible colorings. Then the principle of inclusion-exclusion [23] gives the number of colorings that are not in any of the E_k, i.e., the number of solutions:

$$N = \sum_{i=0}^{e} (-1)^i S_i \tag{4}$$

Each edge eliminates b^{n-1} colorings, so $S_1 = \sum_{l=1}^{e} b^{n-1} = eb^{n-1}$. The colorings that are eliminated by both of two edges require that two nodes have the same color and that either another pair have the same color or a third node has the same color as the first two. In both cases we have b^{n-2} eliminated colorings, giving $S_2 = \binom{e}{2} b^{n-2}$. Thus we see that the first terms in the inclusion-exclusion expansion for the number of solutions depend only on the number of edges in the graph. With three edges, we can have from three to six nodes involved. When there are only three nodes (a triangle), the third edge does not give any additional pruning, i.e., there are b^{n-2} colorings pruned by all three edges. Otherwise, each edges adds a further restriction pruning only b^{n-3}. Thus we have $S_3 = \binom{e}{3} b^{n-3} + (b^{n-2} - b^{n-3})t$ where t is the number of triangles in the graph. Further terms in Eq. 4 involve more complex subgraphs, e.g., squares.

This suggests that the number of triangles in a graph gives a finer determination of classes of critically constrained graphs than just knowing the number of edges. That this is in fact the case is shown in Fig. 4. In particular, we see that there is a transition from mostly solvable to mostly unsolvable cases depending on the number of triangles.

6: Characterizing search methods

The second reason for high variance is the search method itself. To characterize the search heuristic, we return to the simple theory of Eq. 1. A search is difficult when many partial colorings that do not lead to solutions must be considered. A measure of the effectiveness of a heuristic method, i.e., its ability to prune unproductive nodes, can be simply quantified as the probability p that

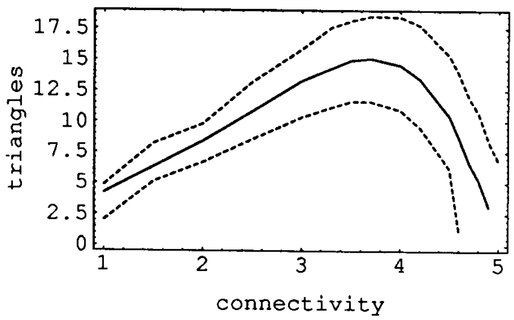

Fig. 4. Fraction of cases with a solution as a function of number of triangles in the graph and γ for $n = 50$, $b = 3$. This was obtained by generating random graphs at γ values in increments of 0.5 and collecting the results based on the number of triangles in each graph. The solid curve shows where half the graphs have a solution, while the upper and lower dashed curves show where fractions 0.2 and 0.8 have solutions, respectively.

an unproductive partial coloring will be recognized as such in the search[1]. A partial coloring of size k is considered only if the smaller colorings preceding it in the search order did not eliminate it, i.e., with probability $(1-p)^{k-1}$ if the pruning ability is independent for each sequence of possible colorings.

The result of this pruning is most simply illustrated when there is no solution. In such a case the expected overall search cost is $C = N_0 + \sum_{k=1}^{n} (1-p)^{k-1} N_k$, whose behavior is shown in Fig. 5. Note first that the cost decreases rapidly as the heuristic is improved. A more subtle observation is the change in behavior as the problem size increases: for poor heuristics, i.e., small p, the cost grows exponentially fast whereas for large p, the cost is roughly independent of problem size.

This behavior is conceptually straightforward. If the heuristic can prune unproductive colorings faster than their number grows with size k, the search will involve fairly limited backtracking giving a low cost. This will be the case when $p > \frac{1}{b}$ so on average each unproductive partial coloring of size k produces less than one additional coloring, of size $k+1$, to include in the search. A less powerful heuristic results in a search involving considerable backtracking and an exponentially growing cost. We thus see another transition behavior, this time due to changes in search method rather than the underlying structure of the problem. This transition behavior can

[1] In practice, such a parameter can be estimated by sampling [3].

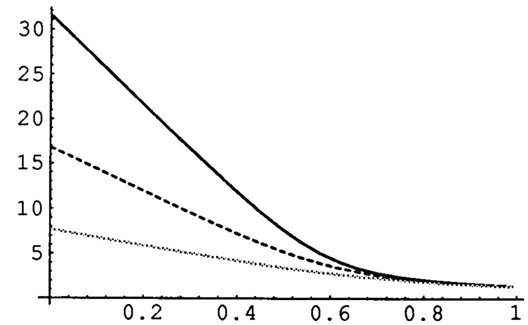

Fig. 5. Behavior of search cost when there are no solutions as a function of heuristic pruning effectiveness. The plot shows $\ln C$ vs. p for n of 20 (gray), 50 (dashed) and 100 (solid), with $\gamma = 6$ so there are unlikely to be any solutions.

also be found in more elaborate models, e.g., including the possibility that the heuristic incorrectly prunes a partial coloring that does lead to a solution[2] as well as cases where there is a solution [13].

7: Generalizations and applications

In summary, we gave a description of the behavior of combinatorial search using graph coloring as an illustration. Unlike the usual case in statistical physics, the large remaining variance prevents accurate predictions of individual cases. To partially address this difficulty, we introduced an additional parameter which refines the location of critically constrained problem instances. We also described a characterization of heuristic search pruning. This study provides an interesting contrast with statistical analyses of physical lattice structures: search is a dynamic process on an exponentially large space which leads to much larger variance than typical in physical systems.

If these behaviors applied only to coloring random graphs, or only to the particular heuristic search used here, they would be of limited interest. However, these transitions have been commonly reported for a variety of combinatorial search problems with a range of very different search methods, as mentioned in the introduction. Such complexity transitions arise not only for constraint satisfaction problems, such as graph coloring, but also for other cases such as optimization [32].

The additional parameter for problem structure generalizes to other constraint problems based on the inclusion-exclusion result for the number of solutions. These problems can all be viewed as due to local inconsistencies

[2] In this case there is the possibility that the search will incorrectly conclude there are no solutions when in fact there are.

from the constraints combining to determine which complete states satisfy all the constraints [31]. In this case the additional parameter relates to the degree of overlap among the local inconsistencies. More accurate theoretical values for the transition points are possible [30] by going beyond the "mean-field" theory of Eq. 1.

An important open issue is the systematic structure of practical combinatorial search problems, and whether this significantly changes the phenomena reported here based on random ensembles. As a partial answer, similar behavior is seen for more restricted classes, such as for those graphs that have a coloring, are connected or where certain "trivial" cases are removed, although the quantitative details change slightly. It remains to be seen whether this continues to be the case for graphs that incorporate additional structure found in particular applications [15, 19] or the ultrametric topology [10] of hierarchical organizations and modularly designed artifacts. Thus, while constraints may be local in the sense of involving only a few aspects of the problem, the induced structure can differ greatly from the locality implied by a euclidean metric in physical systems.

Finally, there are applications of these results to the design of better search algorithms. For example, the theory gives a domain-independent heuristic to identify subproblems that are likely to be particularly hard or easy, on average. This can be used to improve genetic algorithms [4] and may also suggest better choices for backtracking. Knowledge of the likely difficulty of search problems can also be used to determine when a diverse set of methods, perhaps running in parallel and sharing partial results, is most useful [11, 6]. In these cases, improved understanding of the location and nature of the phase transitions in combinatorial search problems can translate directly into more effective search methods.

Acknowledgments

M. Lomelin, A. Sahai and C. Williams contributed to many helpful discussions.

References

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

[2] Peter Cheeseman, Bob Kanefsky, and William M. Taylor. Computational complexity and phase transitions. In *Proc. of the Physics of Computation Workshop*. IEEE Computer Society, October 2-4 1992.

[3] Pang C. Chen. Heuristic sampling: A method for predicting the performance of tree searching programs. *SIAM Journal of Computing*, 21(2):295–315, April 1992.

[4] Scott H. Clearwater and Tad Hogg. Exploiting problem structure in genetic algorithms. In *Proc. of the 12th Natl. Conf. on Artificial Intelligence (AAAI94)*, Menlo Park, CA, 1994. AAAI Press.

[5] James M. Crawford and Larry D. Auton. Experimental results on the cross-over point in satisfiability problems. In *Proc. of the 11th Natl. Conf. on Artificial Intelligence (AAAI93)*, pages 21–27, Menlo Park, CA, 1993. AAAI Press.

[6] Pedro S. de Souza and Saroush Talukdar. Asynchronous organizations for multi-algorithm problems. In *Proc. of ACM Symposium on Applied Computing (SAC93)*, Feb. 1993.

[7] M. E. Dyer and A. M. Frieze. The solution of some random NP-hard problems in polynomial expected time. *J. of Algorithms*, 10:451–489, 1989.

[8] M. R. Garey and D. S. Johnson. *A Guide to the Theory of NP-Completeness*. W. H. Freeeman, San Francisco, 1979.

[9] Ian P. Gent and Toby Walsh. Easy problems are sometimes hard. Research Paper 642, Dept. of AI, Edinburgh Univ., June 27 1993.

[10] J. A. Hartigan. Representation of similarity matrices by trees. *Journal of the American Statistical Association*, 62:1140–1158, 1967.

[11] Tad Hogg and Colin P. Williams. Solving the really hard problems with cooperative search. In *Proc. of the 11th Natl. Conf. on Artificial Intelligence (AAAI93)*, pages 231–236, Menlo Park, CA, 1993. AAAI Press.

[12] Tad Hogg and Colin P. Williams. The hardest constraint problems: A double phase transition. *Artificial Intelligence*, 1994. to appear.

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

[14] 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.

[15] Mark T. Jones and Paul E. Plassmann. A parallel graph coloring heuristic. *SIAM J. Sci. Comput.*, 14(3):654–669, 1993.

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

[17] Scott Kirkpatrick and Bart Selman. Critical behavior in the satisfiability of random boolean expressions. *Science*, 264:1297–1301, 1994.

[18] Tracy Larrabee and Yumi Tsuji. Evidence for a satisfiability threshold for random 3CNF formulas. In Haym Hirsh et al., editors, *AAAI Spring Symposium on AI and NP-Hard Problems*, pages 112–118. AAAI, 1993.

[19] Gary Lewandowski and Anne Condon. Experiments with parallel graph coloring heuristics. In *Proc. of 2nd DIMACS Challenge*, 1993.

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

[21] 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.

[22] 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.

[23] E. M. Palmer. *Graphical Evolution - an Introduction to the Theory of Random Graphs*. Wiley Interscience, NY, 1985.

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

[25] Patrick Prosser. An empirical study of phase transitions in binary constraint satisfaction problems. Technical Report AISL-49-93, Dept. of Computer Science, Univ. of Strathclyde, Glasgow G1 1XH, Scotland, 1993.

[26] 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.

[27] Barbara M. Smith. The phase transition in constraint satisfaction problems: A closer look at the mushy region. Technical Report 93.41, Division of AI, Univ. of Leeds, Leeds LS2 9JT, U.K., 1994.

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

[29] 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.

[30] Colin P. Williams and Tad Hogg. Extending deep structure. In *Proc. of the 11th Natl. Conf. on Artificial Intelligence (AAAI93)*, pages 152–157, Menlo Park, CA, 1993. AAAI Press.

[31] Colin P. Williams and Tad Hogg. Exploiting the deep structure of constraint problems. *Artificial Intelligence*, 1994. to appear.

[32] Weixiong Zhang and Richard E. Korf. An average-case analysis of branch-and-bound with applications: Summary of results. In *Proc. of the 10th Natl. Conf. on Artificial Intelligence (AAAI92)*, pages 545–550, Menlo Park, CA, 1992. AAAI Press.