The Constrainedness Knife-Edge

Toby Walsh
APES Group
Department of Computer Science
University of Strathclyde
Glasgow, Scotland
tw@cs.strath.ac.uk

Abstract

A general rule of thumb is to tackle the hardest part of a search problem first. Many heuristics therefore try to branch on the most constrained variable. To test their effectiveness at this, we measure the constrainedness of a problem during search. We run experiments in several different domains, using both random and non-random problems. In each case, we observe a constrainedness "knife-edge" in which critically constrained problems tend to remain critically constrained. We show that this knife-edge is predicted by a theoretical lower-bound calculation. We also observe a very simple scaling with problem size for various properties measured during search including the ratio of clauses to variables, and the average clause size. Finally, we use this picture of search to propose some branching heuristics for propositional satisfiability.

Introduction

Empirical studies of search procedures usually focus on statistics like the run-time or the total number of nodes visited. It can also be productive to use the computer as a "microscope", looking closely at the running of the search procedure. To illustrate this approach, we measure the constrainedness of problems during search. A general purpose heuristic in many domains is to branch on the most constrained variable. For example, in graph coloring, the Brelaz heuristic colors a node with the fewest available colors, tie-breaking on the number of uncolored neighbours (Brelaz 1979). How effective are heuristics at identifying the most constrained variable? How constrained are the resulting subproblems? To answer such questions, we measured the constrainedness of problems during search in several different domains using both random and non-random problems.

We obtained similar results with a wide variety of algorithms and heuristics. In each case, we observed a constrainedness "knife-edge". Under-constrained problems tend to become less constrained as search deepens, over-constrained problems tend to become more constrained, but critically constrained problems from the region inbetween tend to *remain* critically constrained. We also observe a simple scaling with problem size for various properties measured during search including the ratio of clauses to variables, and the average clause size. The existence of a constrainedness knife-edge helps to explain the hardness of problems from the phase transition. It also suggests some branching heuristics for propositional satisfiability. Similar microscopic studies that look closely inside search may be useful in other domains.

Constrainedness within satisfiability

There has been considerable interest recently in encoding problems into satisfiability and solving them either with local search procedures like GSAT (Selman, Levesque, & Mitchell 1992) or with the Davis-Putnam decision procedure (Bayardo & Schrag 1997). We therefore began our experiments by looking at how the constrainedness of satisfiability problems varies during search. The constrainedness of a satisfiability problem depends on several factors including the clause length (longer clauses are less constraining than shorter ones) and the number of clauses mentioning a variable (increasing the number of clauses makes the variable more constrained). We decided therefore to measure both the ratio of clauses to variables, and the average clause length during search for the popular random 3-SAT problem class (Mitchell, Selman, & Levesque 1992).

We use the Davis-Putnam procedure with unit propagation but no pure literal deletion. We branch with MOM's heuristic, picking the literal that occurs most often in the minimal size clauses. Depth is measured by the number of assignments. Similar results are obtained when depth is measured by the number of branch points, and with other branching heuristics including random branching. In each experiment, we simply follow the heuristic down the first branch, averaging over 1000 different problems. To reduce variance, we use the same ensemble of problems in all experiments. We adopt the convention that initial parameters are in capital italics and that values measured during search are in lower case italics.

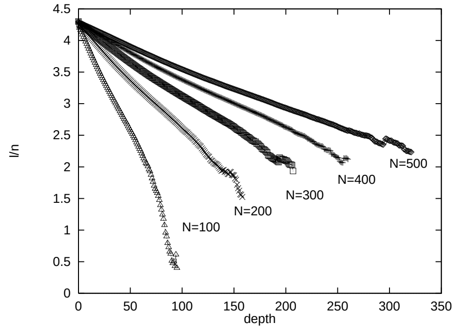

Figure 1: Ratio of clauses to variables, l/n on the heuristic branch against the depth.

In Figure 1, we plot the ratio of clauses to variables down the heuristic branch for random 3-SAT problems from the middle of the phase transition with an initial clause to variable ratio, $L/N = 4.3$. As search progresses, this ratio drops approximately linearly. However, it drops less rapidly for larger problems. Since not all heuristic branches extend to large depths, there is some noise at the end of each graph. Other experiments show that the rate of decay of l/n increases as we increase the initial ratio of clauses to variables, L/N. In Figure 2, we rescale the x-axis linearly with problem size, N. This rescaling shows that the gradient of l/n is inversely proportional to N. Such a simple scaling result is very unexpected. It may be useful in a theoretical analysis of the Davis Putnam procedure.

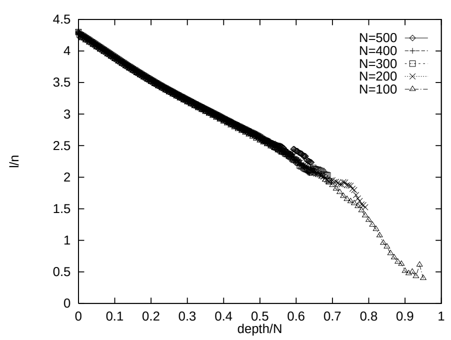

Figure 2: Ratio of clauses to variables, l/n on the heuristic branch against the fractional depth.

As the ratio of clauses to variables drops during search, we might expect that problems become less constrained. However, the average clause length also decreases as search deepens, tightening the constraints on variables. In Figure 3, we show that, just like the ratio of clauses to variables, the average clause length is invariant if depths are scaled linearly with problem size, N. This simple scaling result may also be useful in a theoretical analysis of the Davis Putnam procedure. Other experiments show that the average clause length decreases as we decrease the initial ratio of clauses to variables, L/N. Which of these two factors wins? Does the decrease in clause size tighten the constrainedness faster than the decrease in the ratio of clauses to variables loosens it? To answer such questions, we need a more precise measure of constrainedness.

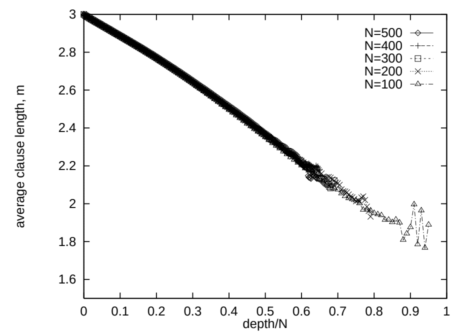

Figure 3: Average clause length, m on the heuristic branch against the fractional depth.

An approximate theory

(Gent *et al.* 1996) proposes an approximate theory for estimating the constrainedness of an ensemble of problems. This theory focuses on just two factors: the size of the problems, and the expected number of solutions. Problems which are large but which have a small number of solutions tend to be over-constrained. On the other hand, problems which are small but which have a large number of solutions tend to be under-constrained. Whilst this theory ignores important factors like problem structure and symmetries, its predictions are often surprisingly accurate. For instance, the theory predicts the location of a phase transition in number partitioning with just a 4% error (Gent & Walsh 1996).

If each problem in an ensemble has a state space with 2^N states, of which $\langle Sol \rangle$ are expected to be solutions, then the constrainedness, κ of the ensemble is defined by,

$$\kappa \quad =_{\text{def}} \quad 1 - \frac{\log_2(\langle Sol \rangle)}{N}$$

This parameter lies within the interval $[0, \infty)$. If $\kappa = 0$, problems in the ensemble are completely under-constrained and every state is a solution. If $\kappa = \infty$, problems in the ensemble are completely over-constrained and no states are solutions. If $\kappa < 1$, problems are under-constrained and are typically soluble. If

$\kappa > 1$, problems are over-constrained and are typically soluble. Around $\kappa \approx 1$, there tends to be a phase transition as problems can be both soluble and insoluble. The hardest problems to solve often occur around such transitions (Cheeseman, Kanefsky, & Taylor 1991).

Constrainedness knife-edge

We can use this definition of constrainedness to determine whether the decrease in average clause size outweighs the decrease in the ratio of clauses to variables. To estimate κ during search, we assume that the current subproblem is taken from an ensemble in which problems have the same number of clauses, the same number of variables, and the same distribution of clause lengths. If there are l_i clauses of length i, then as each clause of length i rules out the fraction $(1 - \frac{1}{2^i})$ of the 2^n possible truth assignments,

$$\langle Sol \rangle \quad \approx \quad 2^n . \prod_i (1 - \frac{1}{2^i})^{l_i}$$

Hence,

$$\kappa \quad \approx \quad - \sum_i \frac{l_i}{n} . \log_2 (1 - \frac{1}{2^i})$$

Note that for a random 3-SAT problem, κ is directly proportional to L/N, the ratio of clauses to variables.

In Figure 4, we plot the estimated constrainedness down the heuristic branch for random 3-SAT problems. For $L/N < 4.3$, problems are under-constrained and soluble. As search progresses, κ decreases as problems become more under-constrained and obviously soluble. For $L/N > 4.3$, problems are over-constrained and insoluble. As search progresses, κ increases as problems become more over-constrained and obviously insoluble. At $L/N \approx 4.3$ problems are on the *knife-edge* between solubility and insolubility. As search progresses, κ is roughly constant. Each successive branching decision gives a subproblem which has the same constrainedness as the original problem, neither more obviously satisfiable, nor more obviously unsatisfiable. Only deep in search does κ eventually break one way or the other.

As with the ratio of clauses to variables, and the average clause length, graphs of the constrainedness during search coincide if depths are scaled linearly with problem size, N. We have also observed similar knife-edge behaviour with a random heuristic, and with an anti-heuristic (that is, one which always branching against the heuristic) except that values of κ are slightly greater.

Figure 4 suggests an interesting analogy with statistical mechanics. At the phase boundary in physical systems, problems tend to be "self-similar". That is,

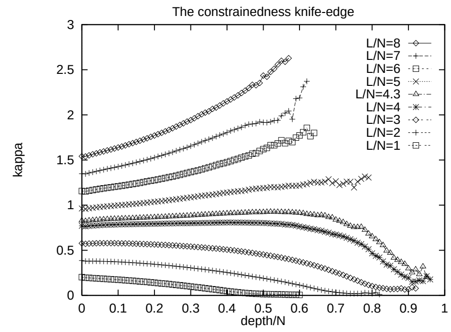

Figure 4: The estimated constrainedness, κ down the heuristic branch for random 3-SAT problems with 100 variables and varying initial ratio of clauses to variable.

they look similar at every length scale. At the phase boundary in computational systems, problems also display a form of self-similarity. Branching decisions give subproblems that look neither more or less constrained. This helps to explain why such problems are difficult to solve. Branching decisions tell us very little about the problem, giving subproblems that are neither more obviously soluble nor more obviously insoluble. We will often have to search to a large depth either for a solution or for a refutation. By comparison, branching on an over-constrained problem gives a subproblem that is often even more constrained and hopefully easier to show insoluble, whilst branching on an under-constrained problem gives a subproblem that is ofen even less constrained and hopefully easier to solve.

Lower bound on constrainedness

When we branch into a subproblem, the number of solutions remaining cannot increase. The expected number of solutions, $\langle Sol \rangle$ cannot therefore increase. This provides a lower bound on κ that is a good qualitative estimate for how the constrainedness actually varies during search. Let κ_i be the value of κ at depth i. Then,

$$\kappa_0 \quad = \quad 1 - \frac{\log_2(\langle Sol \rangle)}{N}$$

Hence,

$$\log_2(\langle Sol \rangle) \quad = \quad N(1 - \kappa_0)$$

Thus,

$$\begin{aligned} \kappa_i \quad &\geq \quad 1 - \frac{\log_2(\langle Sol \rangle)}{N - i} \\ &= \quad 1 - \frac{N(1 - \kappa_0)}{N - i} \\ &= \quad \frac{N\kappa_0 - i}{N - i} \end{aligned}$$

We can improve this bound slightly by noting that κ is bounded below by zero. Hence,

$$\kappa_i \quad \geq \quad \max(0, \frac{N\kappa_0 - i}{N - i})$$

In Figure 5, we plot this bound on κ for random 3-SAT problems with 100 variables and varying initial ratio of clauses to variable, L/N. We see that the behaviour of κ during search observed in Figure 4 is similar to that predicted by the bound.

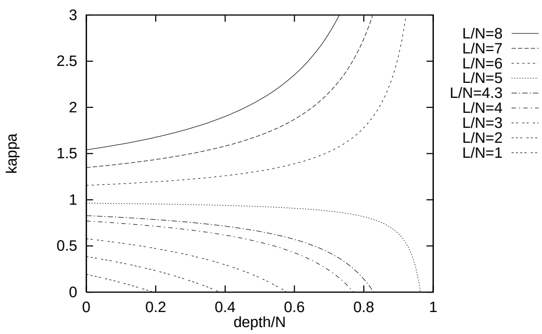

Figure 5: Lower-bound on the constrainedness, κ down a branch for random 3-SAT problems with 100 variables and varying initial ratio of clauses to variable.

Non-random problems

The existence of a constrainedness knife-edge helps to explain the difficulty of solving *random* problems at the phase transition in solubility. Branching decisions give subproblems which are neither more obviously soluble or insoluble. We are forced therefore to search to a large depth either for a solution or for a refutation. Phase transition behaviour has also been observed in problems which are not purely random. For instance, (Gent & Walsh 1995) identifies phase transition behaviour in traveling salesperson problems using real geographical data, in graph coloring problems derived from university exam time-tables, and in Boolean induction and synthesis problem. As a fourth example, (Gomes & Selman 1997) demonstrate phase transition behaviour in the quasi-group completion problem. Does the existence of a constrainedness knife-edge help to explain the difficulty of solving problems at the phase boundary in such non-random problems?

To answer this question, we ran some experiments with graph coloring problems from the DIMACS benchmark library. We used the register allocation problems as these are based on real code. To color the graphs, we use a forward checking algorithm with the Brelaz heuristic to pick the next node to color (Brelaz 1979), and Geelen's promise heuristic to choose one of the m possible colors (Geelen 1992). To estimate κ, we assume that the graph is drawn from an ensemble in which graphs have the same number of nodes, the same available colors, and the same number of edges as in the current subproblem. If V is the set of uncolored nodes, E is the set of edges between uncolored nodes, and m_i is the set of colors remaining for node i then there are $\prod_{i \in V} |m_i|$ possible colorings of the nodes, and each edge $\langle i, j \rangle \in E$ rules out $|m_i \cap m_j|$ of the $|m_i|.|m_j|$ pairs of colors between nodes, i and j. Thus,

$$N \quad = \quad \log_2(\prod_{i \in V} |m_i|) \quad = \quad \sum_{i \in V} \log_2(|m_i|)$$

$$\langle Sol \rangle \quad \approx \quad \prod_{i \in V} |m_i|. \prod_{\langle i,j \rangle \in E} (1 - \frac{|m_i \cap m_j|}{|m_i|.|m_j|})$$

Hence,

$$\kappa \quad \approx \quad -\frac{\sum_{\langle i,j \rangle \in E} \log_2(1 - \frac{|m_i \cap m_j|}{|m_i|.|m_j|})}{\sum_{i \in V} \log_2(|m_i|)}$$

In Figure 6, we plot the estimated constrainedness down the heuristic branch for a typical register allocation problem. Despite the fact that this plot is for a *single* problem instance, we observe a "knife-edge". With less than 49 colors, the problem is over-constrained and insoluble. As search progresses, the constrainedness increases rapidly. Each branching decision results in a subproblem that is more obviously insoluble. With more than 49 colors, the problem is under-constrained and soluble. As search progresses, the constrainedness only increases slightly.

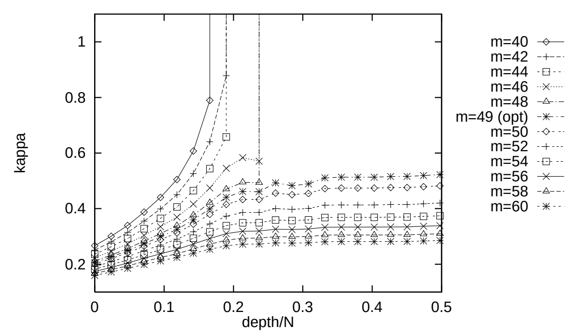

Figure 6: Estimated constrainedness down the heuristic branch for a typical register allocation problem from the DIMACS library using a forward checking algorithm. The problem instance ("zeroin.i.1.col") has 211 nodes, 4100 edges, and needs $m = 49$ colors. For $m \leq 48$, the estimate for κ becomes infinite before the end of search as the problem becomes arc inconsistent.

Each branching decision gives a subproblem that is of similar constrainedness and difficulty to solve. Similar behaviour is seen with the other register allocation problems in the DIMACS library.

Constrainedness within optimization

Phase transition behaviour is not restricted to decision problems like propositional satisfiability. Certain optimization problems like number partitioning and the traveling salesperson problem also exhibit phase transitions (Gent & Walsh 1996; Zhang & Korf 1996). Do we observe a constrainedness knife-edge when solving such optimization problems?

To explore this question, we ran some experiments with the CKK optimization procedure for number partitioning (Korf 1995). Given a bag of N number, we wish to find a partition into two bags that minimizes Δ, the difference between the sum of the two bag. (Gent & Walsh 1996) shows that for partitioning n numbers drawn uniformly at random from $(0, l]$, $\kappa \approx log_2(l)/n$. To estimate κ during search, we assume that the numbers left are taken from such an ensemble and that their size, l is twice the sample average. In Figure 7, we plot this estimate for the constrainedness during search. For comparison, we also plot the lower bound on κ using the same scales. We again observe a constrainedness knife-edge. Although there is not a transition between soluble and insoluble problems (since there is always an optimal partition), there is now a transition between optimization problems with perfect partitions (that is, in which $\Delta \leq 1$) and those without, and verifying the optimality of a partition with $\Delta > 1$ can be costly.

Constrainedness as a heuristic

Knowledge about the existence of a constrainedness knife-edge may help us design more effective search procedures. For instance, for soluble problems, it suggests that we should try to get off the knife-edge as quickly as possible by branching into the subproblem that is as under-constrained as possible. That is, as suggested in (Gent *et al.* 1996), we should branch into the subproblem that minimizes κ. To test this thesis, we implemented a branching heuristic for the Davis-Putnam procedure that branches on the literal which gives the subproblem with smallest κ. In Table 1, we show that this heuristic performs well on hard and satisfiable random 3-SAT problems.

For insoluble problems, the existence of a constrainedness knife-edge suggests that we should branch into the sub-problem that is as over-constrained as possible. That is, we should branch into the subproblem that maximizes κ. Initial experiments suggest

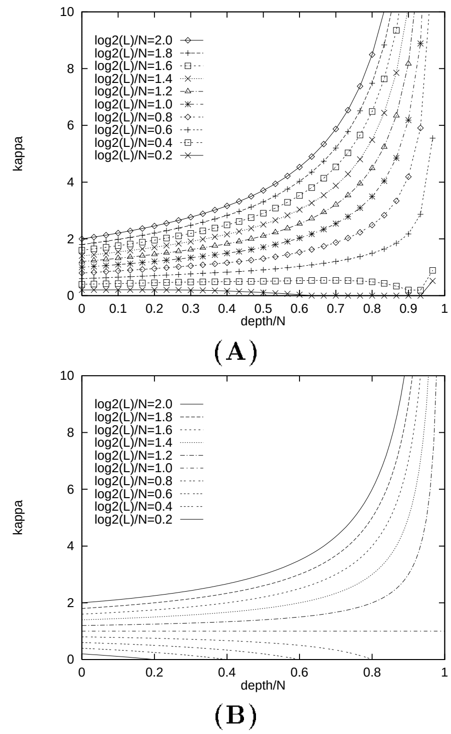

Figure 7: The constrainedness, κ down the heuristic branch for number partitioning problems with $N = 30$ numbers, and varying L. **(A)** estimated κ. **(B)** theoretical lower-bound to same scale.

that this heuristic is effective on hard and unsatisfiable random 3-SAT problems. For instance, for 50 variable unsatisfiable problems at $L/N = 4.3$, the median nodes searched using this heuristic is 2,575 compared to 3,331 nodes for MOM's heuristic, and 7,419 nodes for the heuristic that minimizes κ. On the other hand, maximizing κ is less effective on hard and satisfiable problems. For 50 variable satisfiable problems at $L/N = 4.3$, the median nodes searched when maximizing κ is 1,487 compared to 164 nodes with MOM's heuristic, and 104 nodes with the heuristic that minimizes κ. An adaptive heuristic that switches between minimizing and maximizing κ depending on an estimate of the solubility of the problem may therefore offer good performance.

Related work

Most theoretical studies of the Davis-Putnam procedure have used the easier constant probability model. One notable exception is (Yugami 1995) which computes the average-case complexity of the Davis-Putnam procedure for the random 3-SAT problem class. Freeman has studied experimentally the running of the

N	MOM	KAPPA
25	11	1
50	164	104
75	1129	580
100	3903	1174

Table 1: Median nodes searched by the Davis-Putnam procedure for satisfiable random 3-SAT problems at $L/N = 4.3$, branching either with MOM's heuristic, or to minimize the constrainedness (KAPPA).

Davis-Putnam procedure on random 3-SAT problems (Freeman 1996). Unlike here, where the focus is on the heuristic branch, Freeman computes averages across all branches in the search tree. He identifies an "unit cascade", a depth in the search tree where unit propagation greatly simplifies the problem. The ineffectiveness of unit propagation above this depth helps to explain the hardness of problems at the phase transition.

Gent and Walsh have studied experimentally the running of local search procedures for satisfiability (Gent & Walsh 1993). They show that various properties like the percentage of clauses satisfied, and the number of variables offered to flip are invariant if depths are scaled linearly with problem size. This mirrors the result here on the scaling of the constrainedness, the ratio of clauses to variables and the average clause size. Such simple scaling results may be useful in the theoretical analysis of these search procedures.

Conclusions

We have measured how the constrainedness of problems varies during search in several different problem domains: both decision problems like propositional satisfiability and graph coloring, and optimization problems like number partitioning. Our experiments have used both random and non-random problems. In each case, we observed a constrainedness "knife-edge" in which critically constrained problems tended to remain critically constrained. The existence of a constrainedness knife-edge helps to explain the hardness of problems from the phase transition. We have shown that a lower-bound calculation predicts this knife-edge theoretically. We have also observed a very simple scaling with problem size for various properties measured during search like the constrainedness, the ratio of clauses to variables, and the average clause size. Finally, we have used the existence of a constrainedness knife-edge to propose some branching heuristics for propositional satisfiability. We conjecture that similar microscopic studies that look closely inside search may be useful in other domains.

Acknowledgments

I wish to thank members of the APES group, especially Barbara Smith and Ian Gent, for their comments and criticisms. The author is supported by EPSRC award GR/K/65706.

References

Bayardo, R., and Schrag, R. 1997. Using CSP look-back techniques to solve real-world SAT instances. In *Proceedings of the 14th National Conference on AI.*

Brelaz, D. 1979. New methods to color the vertices of a graph. *Commincations of ACM* 22:251–256.

Cheeseman, P.; Kanefsky, B.; and Taylor, W. 1991. Where the really hard problems are. In *Proceedings of the 12th IJCAI*, 331–337.

Freeman, J. 1996. Hard random 3-SAT problems and the Davis-Putnam procedure. *Artificial Intelligence* 81(1-2):183–198.

Geelen, P. 1992. Dual viewpoint heuristics for binary constraint satisfaction problems. In *Proceedings of the 10th ECAI*, 31–35.

Gent, I., and Walsh, T. 1993. An empirical analysis of search in GSAT. *Journal of Artificial Intelligence Research* 1:23–57.

Gent, I., and Walsh, T. 1995. Phase transitions from real computational problems. In *Proceedings of the 8th Int. Symp. on Artificial Intelligence*, 356–364.

Gent, I., and Walsh, T. 1996. Phase transitions and annealed theories: Number partitioning as a case study. In *Proceedings of ECAI-96.*

Gent, I.; MacIntyre, E.; Prosser, P.; and Walsh, T. 1996. The constrainedness of search. In *Proceedings of the 13th National Conference on AI.*

Gomes, C., and Selman, B. 1997. Problem structure in the presence of perturbations. In *Proceedings of the 14th National Conference on AI*, 221–226.

Korf, R. 1995. From approximate to optimal solutions: A case study of number partitioning. In *Proceedings of the 14th IJCAI.*

Mitchell, D.; Selman, B.; and Levesque, H. 1992. Hard and Easy Distributions of SAT Problems. In *Proceedings of the 10th National Conference on AI*, 459–465.

Selman, B.; Levesque, H.; and Mitchell, D. 1992. A New Method for Solving Hard Satisfiability Problems. In *Proceedings of the 10th National Conference on AI*, 440–446.

Yugami, N. 1995. Theoretical analysis of Davis-Putnam procedure and propositional satisfiability. In *Proceedings of the 14th IJCAI*, 282–288.

Zhang, W., and Korf, R. 1996. A study of complexity transitions on the asymmetic traveling salesman problem. *Artificial Intelligence* 81(1-2):223–239.