The Dynamics of Dynamic Variable Ordering Heuristics

Patrick Prosser

Department of Computer Science, University of Strathclyde, Glasgow G1 1XH, Scotland. E-mail: pat@cs.strath.ac.uk

Abstract. It has long been accepted that dynamic variable ordering heuristics outperform static orderings. But just how dynamic are dynamic variable ordering heuristics? This paper examines the behaviour of a number of heuristics, and attempts to measure the entropy of the search process at different depths in the search tree.

1 Introduction

Many studies have shown that dynamic variable ordering (dvo [9]) heuristics out perform static variable ordering heuristics. But just how dynamic are dynamic variable ordering heuristics? This might be important because if we discover that some dvo heuristic H_1 results in less search effort than heuristic H_2 and H_1 is more dynamic than H_2 then we might expect that we can make a further improvement by increasing the dynamism of H_1 . Conversely if we discover that H_1 is better and less dynamic then we might plan to make H_1 even more ponderous. But how do we measure the dynamism of a heuristic? To investigate this we first look inside the search process, and define our measure of entropy. We then measure entropy for a variety of heuristics. A further examination of the search process reveals that the different heuristics have different signatures, distributing their search effort over different depths of the search tree.

2 Inside Search

Tabulated below is the number of selections of each variable at each depth in the search tree, for a single instance of a randomly generated binary csp, $\langle 20, 10, 0.5, 0.37 \rangle^1$, as seen by a forward checking routine with a dynamic variable ordering heuristic. Each row corresponds to a depth in search (20 in all) and each column represents a variable (again, 20 in all, with the first column entry being row/depth number). Looking at row 3 for example we see that variable V_3 was selected 8 times, variable V_7 selected once, V_8 selected 3 times, and so on. A variable V_i is selected at depth d if at depth d - 1 the current variable is consistently instantiated and the next variable selected by the heuristic at depth d is V_i . The data below corresponds to a single soluble instance.

¹ The problem has 20 variables, each with a domain of 10 values. The proportion of constraints in the graph is 0.5, and the proportion of possible pairs of values in conflict across a constraint is 0.37.

[©] Springer-Verlag Berlin Heidelberg 1998

						V	is	it	3 8	at	De	ept	th								Entropy
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0.0
3	0	0	8	0	0	0	1	3	0	0	2	0	0	0	1	1	0	0	1	0	2.28
4	0	7	5	1	0	0	8	3	0	0	1	0	4	0	0	1	1 (0 ();	30	2.85
5	0	2	5	2	2	0	6	4	7	0	3	0	3	0	0	5	0	0	1	1 0	3.24
6	0	3	1	2	0	0	3	2	2	0	2	0	3	0	2	4	0	0	5	1	3.44
7	0	4	0	0	1	1	1	1	0	2	0	0	2	2	0	0	0	0	0	3	2.98
8	0	0	0	0	2	0	1	0	0	3	1	0	1	0	1	0	0	1	0	1	2.85
9	0	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2.0
10	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1.0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0.0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0.0
13	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0.0
14	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0.0
15	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0.0
16	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0
17	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0.0
19	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0
20	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0

The column out to the right is the measured value of *entropy* for the data in that row.

3 Entropy

Entropy is a measure of the disorder within a system, or the information within the system (i.e. the number of bits required to represent that system). If the system is totally ordered, we will require few bits of information to represent the system, and it will have low entropy. If the system is very disordered we will require many bits to describe the system, and it will have high entropy. Therefore, we might measure the entropy resulting from the variable ordering heuristic at each depth in the search tree. If the heuristic is static, always selecting the same variable at a given depth, then entropy will be a minimum. If the heuristic is very dynamic, selecting freely any future variable at a given depth, entropy should be a maximum.

From thermodynamics, entropy is k.log(w) where k is Boltzmann's constant and w is the *disorder* parameter, the probability that the system will stay in its current state rather than any other state. For our application we measure entropy at depth d as

$$\sum_{i=1}^{n} -p_{d,i} . log_2(p_{d,i})$$
(1)

where $p_{d,i}$ is the probability of selecting variable V_i at depth d. Looking at the tabulation above, for the first row d = 1, only one variable is selected at this depth (the root of the search tree) and entropy is zero. At depth d = 2 we see

that only V_{17} is visited, but three times. Again $p_{2,17} = 1$ and entropy is again zero. The third row d = 3, there are 17 visits at this depth, variable V_3 is visited 8 times, consequently $p_{3,3} = 8/17$, $p_{3,7} = 1/17$, $p_{3,8} = 3/17$, and so on. The entropy at depth d = 3 is then

```
-[8/17.log(8/17) + 1/17.log(1/17) + 3/17.log(3/17) +
2/17.log(2/17) + 1/17.log(1/17) + 1/17.log(1/17) +
1/17.log(1/17)]
= 2.28
```

If all the *n* variables are selected the same number of times at depth *d*, then the entropy at that depth is $log_2(n)$, and this is a maximum, the number of bits required to represent the *n* variables selected. Conversely if only 1 variable is ever selected at depth *d* then entropy at that depth is zero (we require no bits to represent this). If a dvo heuristic is highly dynamic at a certain depth we expect a correspondingly high entropy, and if the variable ordering is static we have zero entropy.

4 Entropy at Depth

Experiments were carried out on 100 instances of $\langle 20, 10, 0.5, 0.37 \rangle$ problems (from the crossover point [7]). Of these, 54 were soluble and 46 insoluble. The search algorithm used was forward checking with conflict-directed backjumping (fc-cbj [8]). Five heuristics were investigated:

- FF, fail-first, choosing the variable with smallest current domain, tie breaking randomly [6, 9].
- BZ, Brelaz heuristic, essentially FF tie breaking on the variable with most constraints acting into the future subproblem, and tie breaking further randomly [1].
- GEL, Geelen's combined variable and value ordering heuristic, selecting the most promising value for the least promising variable [2].
- KP, the minimise- κ heuristic, selecting the variable that leaves the future subproblem with the lowest κ value [5].
- RAND, a random selection at each point. When a variable is selected we pick at random from the future variables. RAND is the *straw man* to show just what effect natural dynamism has on entropy at depth.

We might say that as we move from FF to BZ to KP to GEL we move towards more informed heuristics.

Figure 1 shows average entropy at depth (on the left) for the 54 soluble instances, and (on the right) for the 46 insoluble instances. A contour is given for each of the heuristics. The contour for RAND (our straw man) shows that at depths 5 to about 12 entropy is constant at about 4.2, and this corresponds closely to what theory predicts. That is, at depth 1 a variable has been selected and is withdrawn from future selections. Consequently greater depths can select

19



Fig. 1. Entropy at Depth for $\langle 20, 10, 0.5, 0.37 \rangle$ problems; on the left (a) 54 soluble problems, and on the right (b) 46 insoluble problems. Note that the tail of the contours in (b) for RAND, FF, and BZ have relatively small sample sizes.

from at most 19 variables. If each variable is selected at a given depth with equal probability entropy will be $log_2(19) \approx 4.25$, and this is what we observe.

The FF heuristic is significantly different from RAND; entropy is generally lower at all depths, and entropy falls away at a shallower depth. More generally, what we see is less entropic behaviour as heuristics become more informed. This pattern appears to hold, but maybe to a lesser extent over insoluble problems (Figure 1(b)).

5 Effort inside search

We now investigate how effort is distributed across the depths of search. First we tabulate the overall performance of the heuristics, in terms of consistency checks and nodes visited.

	Sol	luble	Insc	Insoluble			
	Checks	Nodes	Checks	Nodes			
RAND	444.8	29.7	1216.7	80.3			
FF	29.1	1.1	68.8	2.7			
ΒZ	15.4	0.6	34.8	1.3			
KP	16.8	0.7	37.8	1.6			
GEL	16.8	0.7	59.6	2.6			

The table above shows for each heuristic the performance measured as the average number of consistency checks (measured in thousands) for the soluble and the insoluble problems, and nodes visited (again in thousands). No claims are drawn from the above results, for example that one heuristic is better than another, because the sample size is too small and the problem data too specific².

² For example, we will get a different ranking of the heuristics if we vary problem features[3].

The contours in Figure 2 show, for the RAND heuristic, the average number of consistency checks performed at varying depths in the search tree, nodes visited, and variables selected. Note that the y-axis is a logscale. The curves look quite natural, with the peak in search effort taking place in the first third of search.



Fig. 2. Average Checks, Nodes Visited and Variables Selected at Depth for $\langle 20, 10, 0.5, 0.37 \rangle$ problems using RAND dvo; on the left (a) 54 soluble problems, and on the right (b) 46 insoluble problems.

Figure 3 shows average consistency checks only, for the four dvo's: FF, BZ, KP, and GEL. The contours are very different from RAND, compressing the search effort into a relatively narrow band at shallow depth. Also note that KP and GEL typically dispense with search after depth 9, thereafter walking to the solution without backtracking. Figure 3 suggests that each heuristic has a different signature. KP and GEL appear to squeeze all the search effort up to a shallow depth, and are reminiscent of the different signatures of forward checking and mac-based algorithms [10].



Fig. 3. Average Checks at Depth for (20, 10, 0.5, 0.37) problems; on the left (a) 54 soluble problems, and on the right (b) 46 insoluble problems.

22 P. Prosser

Figure 4 shows the average number of nodes visited by each of the heuristics (excluding RAND) at various depths. These contours are very similar to those in Figure 3, as expected, showing that consistency checks correlate with visits.



Fig. 4. The average number of nodes visited at depth for the $\langle 20, 10, 0.5, 0.37 \rangle$ problems; on the left (a) 54 soluble problems, and on the right (b) 46 insoluble problems.

6 Conclusion

A small empirical study has been presented, investigating the behaviour of dynamic variable ordering heuristics. We have attempted to measure the dynamism of dvo heuristics using entropy, and it appears that the more informed a heuristic the less entropic/dynamic its behaviour. We also see that the heuristics examined have markedly different signatures, moving the search effort to different depths in the search tree.

Further work should be done, in particular different ways of measuring entropy should be explored. Rather than measure it across depths in the search tree, maybe it can be measured along paths, or maybe just arcs in the search tree. We might also investigate the heuristic signature, and see if we can predict how search effort grows at depths for different heuristics (maybe using finite size scaling [4]). This might then allows us to predict how search cost scales for different heuristics within the search process.

Acknowledgements

I would like to thank my colleagues (past and present) in the APES research group. In particular Craig Brind, Dave Clark, Ian Philip Gent, Stuart Grant, Phil Kilby, Ewan MacIntyre, Andrea Prosser, Paul Shaw, Barbara Smith, Kostas Stergiou, Judith Underwood, and Toby Walsh. I would also like to thank Peter van Beek for encouraging us to ask such interesting questions.

References

- 1. D. Brelaz. New methods to color the vertices of a graph. JACM, 22(4):251-256, 1979.
- P.A. Geelen. Dual viewpoint heuristics for the binary constraint satisfaction problem. In Proc. ECAI92, pages 31-35, 1992.
- I.P. Gent, E. MacIntyre, P. Prosser, B.M. Smith, and T. Walsh. An empirical study of dynamic variable ordering heuristics for constraint satisfaction problems. In Proc. CP96, pages 179-193, 1996.
- I.P. Gent, E. MacIntyre, P. Prosser, and T. Walsh. Scaling effects in the CSP phase transition. In *Principles and Practice of Constraint Programming*, pages 70-87. Springer, 1995.
- I.P. Gent, E. MacIntyre, P. Prosser, and T. Walsh. The constrainedness of search. In Proc. AAAI-96, 1996.
- R.M. Haralick and G.L. Elliott. Increasing tree search efficiency for constraint satisfaction problems. Artificial Intelligence, 14:263-313, 1980.
- T. Hogg, B.A. Huberman, and C.P. Williams. Phase transitions and the search problem (editorial). Artificial Intelligence, 81(1-2):1-15, 1996.
- 8. P. Prosser. Hybrid algorithms for the constraint satisfaction problem. Computational Intelligence, 9(3):268-299, 1993.
- P.W. Purdom. Search rearrangement backtracking and polynomial average time. Artificial Intelligence, 21:117-133, 1983.
- D. Sabin and E.C. Freuder. Contradicting conventional wisdom in constraint satisfaction. In Proc. ECAI-94, pages 125-129, 1994.