Variable Ordering Heuristics Show Promise[⋆]

J. Christopher Beck[*], Patrick Prosser[†], and Richard J. Wallace[*]

[*]Cork Constraint Computation Centre, Department of Computer Science,
University College Cork, Cork, Ireland
{c.beck,e.freuder}@4c.ucc.ie
[†]Department of Computing Science, University of Glasgow, Scotland,
pat@dcs.gla.ac.uk

1 Introduction

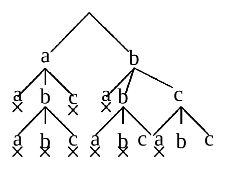
Promise is the ability to make choices that lead to a solution when one exists. The traditional intuition behind variable ordering heuristics is Haralick and Elliott's fail-first principle: choose the variable such that assigning it is most likely to lead to a domain wipe-out (in AIJ 14, 1980). In contrast, the standard belief about value ordering heuristics is based on Geelen's discussion (in ECAI'92): choose a value that is most likely to participate in a solution. It is not clear *a priori* that changes in variable ordering change the likelihood of finding a solution in a way that will affect overall performance significantly. In this paper we show that promise does have a meaning for variable ordering heuristics and that the level of promise of a variable ordering heuristic can be measured. In addition, we show that the promise of different variable ordering heuristics is different and that the level of promise of a variable ordering heuristic correlates with search cost for problems with many solutions.

2 A Measure of Promise

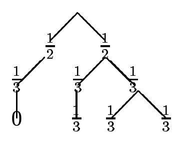
A reasonable approach to measuring promise is to assess the extent to which a variable ordering heuristic increases or decreases the likelihood of finding a solution. It is critical that this measurement is independent of a heuristic's ability to escape bad subtrees, i.e. we must ensure that any measure of promise is not contaminated by the fail-firstness of a heuristic. The "likelihood of finding a solution" can be treated probabilistically. For a given decision there is some probability over all possible subsequent decisions that the choice will lead to a solution. We can also consider promise with respect to an entire problem, i.e. as an expected value over all possible sequences of choices. In this sense we can speak of the promise of a problem in terms of its relation to a perfect selection. This measure of promise has a natural minimum and maximum: 0 (for insoluble problems) and 1 (if every n-tuple is a solution). This gives us a universal measure across all problems. These points can be illustrated with the following toy problem. We have three variables v_1, v_2 and v_3 with domains $d_1 = \{a, b\}$, $d_2 = \{a, b, c\}$, and $d_3 = \{a, b, c\}$, and the constraints $C_{1,2} = \{(a, b), (b, b), (b, c)\}$, $C_{1,3} = \{(a, a), (a, b), (b, b), (b, c)\}$, and $C_{2,3} = \{(a, a), (b, c), (c, a), (c, b), (c, c)\}$. To calculate promise, we consider the

[⋆] This work has received support from Science Foundation Ireland under Grant 00/PI.1/C075 and ILOG, SA.

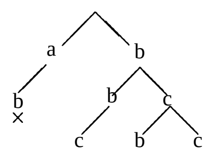
probability of choosing each viable value from a domain, when any value is equally likely to be chosen. For simple backtracking we get the tree in Figure (a), and from this we can calculate the overall promise for backtracking on this problem by summing the path-products (shown in Figure (b)), giving a value of $0 + \frac{1}{18} + \frac{2}{18} = \frac{1}{6}$. The overall promise for this problem and this consistency algorithm is then $\frac{1}{6}$, and for backtracking this is the same as the solution density. Consider forward checking (fc) using the smallest-domain-first (sdf) ordering. The search tree is shown in Figure (c) and we again compute promise by summing the path products in tree (c) giving the sum: $0 + \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$. If we use a different variable ordering, say 3-2-1 instead of 1-2-3, we get a different search tree, Figure (d), and a different measure of promise: $0 + \frac{1}{3} + \frac{1}{6} + \frac{1}{6} = \frac{2}{3}$.



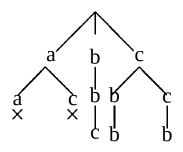
(a) Backtrack Tree



(b) Probability Tree of (a)



(c) Forward Checking Tree, ordered 1-2-3



(d) Forward Checking Tree, ordered 3-2-1

From these examples we now draw some conclusions: (1) Promise can vary depending on the variable ordering and the consistency algorithm. (2) Promise is not in general equivalent to solution density; the equivalence holds only if there is no consistency maintenance. (3) There does not appear to be a particular variable ordering that is guaranteed to maximize promise. Therefore, the level of promise of a variable ordering heuristics will probably have to be decided empirically.

It is possible to assess the overall promise of a problem under a given variable ordering with the following *probing* procedure. Use a heuristic to select an uninstantiated variable, randomly select a value for this variable, and then enforce some level (possibly none) of consistency. Repeat these steps until all variables are consistently instantiated or a deadend is encountered. At a deadend the probing process re-starts from the beginning with a new random seed. This is repeated until a complete solution is found. The number of probes required gives us a measure of promise: the greater the promise the fewer the number of probes required on average to obtain a solution. (In fact, in the limit the expected number of probes is the reciprocal of the promise.)

This procedure avoids contamination by fail-firstness because we never try to recover from a deadend: the number of probes are a reflection of promise alone. If used with a random value ordering heuristic over many runs, this technique also avoids effects of value selection on promise. We can use any consistency enforcement algorithm as part of the probing procedure.

3 Empirical Investigations

We use the probing procedure to investigate two hypotheses: (1) *Different variable ordering heuristics exhibit different levels of promise*, and (2) *Promise is inversely correlated with search effort*. We expect that for easier problems promise will be strongly correlated with search cost and as problems become more difficult, the effect of promise will decrease and fail-firstness should be more important to search effort.

We use a set of well-known variable ordering heuristics together with their corresponding anti-heuristics: sdf (smallest domain first) and ldf (largest domain first); max- and min-static-degree; max- and min-forward-degree; Brélaz heuristic and anti-brelaz (i.e. choose the variable with the smallest (resp. largest) domain and break ties by choosing the variable with maximum (resp. minimum) forward degree (CACM 22, 1979)); domdeg and anti-domdeg (i.e. choose the variable that minimizes (resp. maximizes) the ratio of domain size to forward degree (Bessiere & Régin, CP'96)); and finally random (i.e. randomly select an unassigned variable).

Haralick and Elliott's Forward checking (fc) is used with each heuristic. We conduct the probing procedure 100 times for each problem and variable ordering heuristic, with different seeds for the random number generator. The mean reciprocal of the number of probes over the 100 runs is our estimate of promise for that problem and heuristic. For problems with no solutions, we define the promise to be 0. For a set of problems and a variable ordering heuristic, we calculate promise by finding the arithmetic mean of the promise estimate over each problem in the set for that heuristic.

To estimate the search effort for a problem and a variable ordering heuristic we follow a similar procedure as for estimating promise. The difference is that instead of probing for solutions we simply use a complete backtracking search with forward checking. Our measure of search effort is the number of consistency checks required to find a solution. Again, for a given problem and variable ordering heuristic, we run this algorithm 100 times with differing random seeds for value selection, and define the search cost to be the median number of constraint checks over the 100 runs. For a set of problems the mean search cost is the arithmetic mean of the search cost for each problem.

We first test our hypotheses on randomly generated CSPs. The test problems are generated using the $\langle n, m, p_1, p_2 \rangle$ model of Smith and Grant (ECAI'98), with 15 variables and 10 values per variable. Density p_1 is fixed at 0.7, and tightness p_2 varies from 0.30 to 0.39 in steps of 0.01. There are 100 problems at each value of p_2. Problems are not filtered and therefore samples may contain a mix of soluble and insoluble problems.

The left graph in Figure 1 presents the mean promise estimates for each variable ordering heuristic. A log-scale is used on the y-axis to make the rankings of the heuristics easier to see. The right graph in Figure 1 presents the search cost for each variable ordering heuristic.

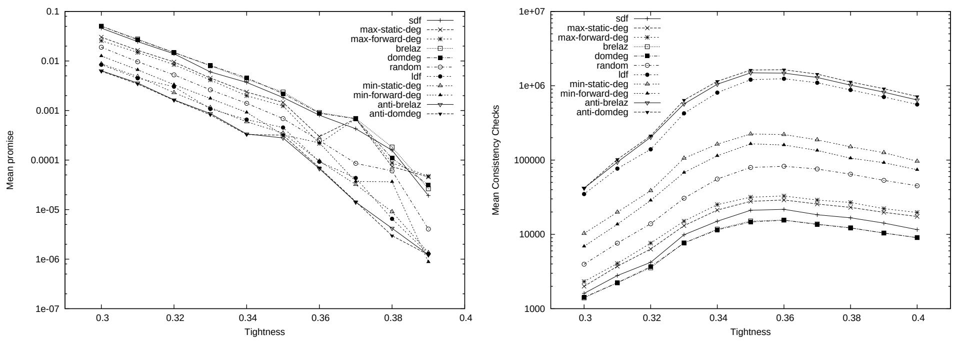


Fig. 1. Mean promise estimates (left) and mean search cost (right) for each random problem set.

The promise graph clearly shows that different variable ordering heuristics do exhibit different levels of promise. As the tightness increases there are fewer soluble problems and the level of promise correspondingly decreases. Even when we remove the insoluble problems (not shown) the promise decreases with increasing tightness as fewer solutions result in a lower promise for a given variable ordering heuristic. Comparing the two plots in Figure 1, we can see that the most successful heuristics (i.e. those with lowest search cost: domdeg, brelaz, and sdf) also exhibit the highest levels of promise. Furthermore, the rankings seem relatively consistent: with some exceptions that may be due to noise from problem sets with few soluble problems, the i^{th} best heuristic exhibits the i^{th} highest level of promise.

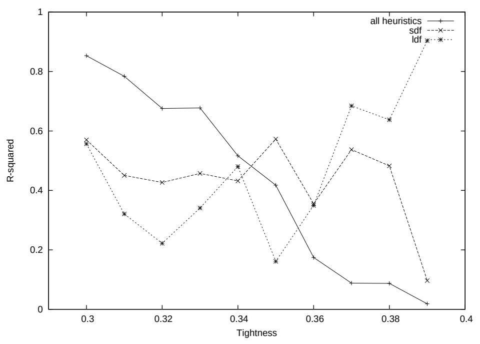


Fig. 2. R^2 values for the correlations between promise and the reciprocal of search cost for each random problem set and selected variable ordering heuristics.

Figure 2 presents a measure of the correlation between promise and search cost. The plot labeled “all” examines 1100 points for each problem set composed of the search cost and promise values for each of the 100 test problems and 11 variable ordering heuristics. For low values of tightness, the variation in promise accounts for 70-80% of the variation in search cost. As the problems become tighter the R^2 value drops to close to 0.

Figure 2 also presents the R^2 values for sdf and its anti-heuristic ldf. This plot is similar to that seen with the other heuristic/anti-heuristic pairs based on domain size (i.e., domdeg and brelaz): at low values of tightness the R^2 values are somewhat noisy and around 0.4. For the tighter problem sets, the anti-heuristic ldf starts to have a much stronger correlation with search cost while the correlation of the heuristic sdf declines. The reason for this effect is that the anti-heuristics perform uniformly poorly across the insoluble problems at a tightness value. Given that there are very few soluble problems at high tightness values (i.e. one for set 0.39) this leads to a strong correlation between promise and search cost. Another surprising result is that for the problem sets with low tightness, the R^2 values for individual heuristics are much lower than the "all" plot. This occurs because the relationship between promise and search cost is weaker within any single heuristic, but over all heuristics the trend of higher promise corresponding to lower search cost for loose problems is clear.

The heuristics and anti-heuristics based only on variable degree have a similar behavior as the domain-based heuristics for low values of tightness. As the tightness increases the R^2 values of both the heuristics and anti-heuristics drop as expected. Lending support to our above explanation of the domain-based anti-heuristics, the degree-based anti-heuristics have large variances on insoluble problems.

4 Observations, Discussion, and Conclusion

Using random CSPs, we have demonstrated a definite inverse relation between promise and search effort across different variable ordering heuristics. We have also shown that promise has a high (inverse) correlation with search effort for problems with many solutions but a low correlation when the number of solutions decreases. There is no question, then, about having to consider promise as well as fail-firstness in trying to account for differences in the performance of these heuristics.

Why should heuristics such as sdf, domdeg, and brelaz show greater degrees of promise than other variable orderings? These heuristics give preference to variables with small domain sizes. If we assume that each value in the domain of a variable has equal probability of occurring in a solution, when domain sizes are small the probability of any value in that domain being in a solution is relatively high. That is, *the values are more promising*. Therefore we should expect that heuristics that prefer variables with small domains will have higher promise, and this is just what we have seen. This explanation may also account for the differences between heuristics and anti-heuristics that are based on degree alone, although this has not yet been tested.

A full understanding of search heuristics will require further work. By focusing on promise, we have neglected the fail-first principle in this paper. The development of a method of measurement of fail-firstness and the investigation of its correlation with search cost is a key area for future work. Our intuition is that it is the combination of promise and fail-firstness that, to a large extent, determines the search efficiency of heuristics.