

Failure Analysis in Backtrack Search for Constraint Satisfaction*

Tudor Hulubei and Barry O’Sullivan

Cork Constraint Computation Centre
Department of Computer Science, University College Cork, Ireland
tudor@hulubei.net, b.osullivan@cs.ucc.ie

1 Introduction

Search effort is typically measured in terms of the number of backtracks, constraint checks, or nodes in the search tree, but measures such as the number of incorrect decisions have also been proposed. Comparisons based on mean and median effort are common. However, other researchers focus on studying runtime distributions, where one can observe a (non-)heavy-tailed distribution under certain conditions [2, 3].

In this paper we augment our traditional statistics-based approach to studying systematic search with a visualisation method that uses heatmaps, which can be more informative and present different views of the relationship between the depth at which a mistake occurred and the size of the refutation associated with it. We compare search algorithms on the basis of the number of, and effort required to recover from, *individual mistakes*. We highlight interesting differences between random and real-world problems, contradicting conventional wisdom that states that mistakes at the top of the search tree are much more expensive to refute than those made deeper in the tree.

We also observe some interesting patterns in terms of where most of the search effort is consumed *over a large population of problem instances* and show that it is not always the case that extremely large mistakes account for most of the effort. Finally, we show that variable ordering heuristics alone can avoid making mistakes, but that their performance cannot be attributed exclusively to either fail-firstness or promise.

2 Experiments

We study the failure characteristics of backtrack search methods in constraint satisfaction problems. Our analysis is based on counting the number of times an assignment was made during search that took us off the path to a solution. We refer to such decisions as *mistakes* [5]. The set of nodes visited by the algorithm in order to recover from a mistake is the *refutation tree* of that mistake. The number of nodes in that tree is the *refutation size*, which is our measure of effort.

Our empirical analysis includes configurations of uniform Model B random binary problems and quasigroup completion problems, encoded using binary constraints. With

* This work was supported by Science Foundation Ireland (Grant 00/PI.1/C075). We thank the Boole Centre For Research in Informatics for providing access to their Beowulf cluster.

the exception of the random 17×8 problems (i.e. 17 variables with uniform domain size 8), where we used backtracking, all other experiments used MAC. Our data sets of random problems contain approximately 10,000 instances for each algorithm used. The QWH-10 data set includes a total of over 1,000,000 instances. Specifically, our data sets comprise of the following problems:

1. Dense random 30×10 , density 0.86, tightness 0.15, at the phase transition.
2. Sparse random 30×10 , density 0.3, tightness 0.35, in the easy region.
3. Sparse random 150×10 , density 0.03356, tightness 0.52, in the easy region.
4. QWH-10 with 90% random balanced holes.
5. Random 17×8 , density 0.84, tightness 0.09375 (easy but heavy-tailed when using random orderings); and 0.25, near the phase transition and non-heavy-tailed.

Rather than using the overall effort required to solve each instance, which we refer to as *instance-based effort*, as the basis of our analysis, we also considered the effort required to refute each mistake separately, which we refer to as *mistake-based effort*. We established that they were highly correlated, and so observations made on the latter can be used to draw conclusions about the former. Studying search algorithms at the mistake-level allows us to perform a more detailed analysis of the interactions between variable and value ordering heuristics over a large population of instances. For the remainder of the paper we will base our comparison of search heuristics on an analysis of mistake-level effort through the use of heatmaps (best viewed in colour at <http://hulubei.net/tudor/papers/fabscs>).

2.1 Distribution of Search Effort

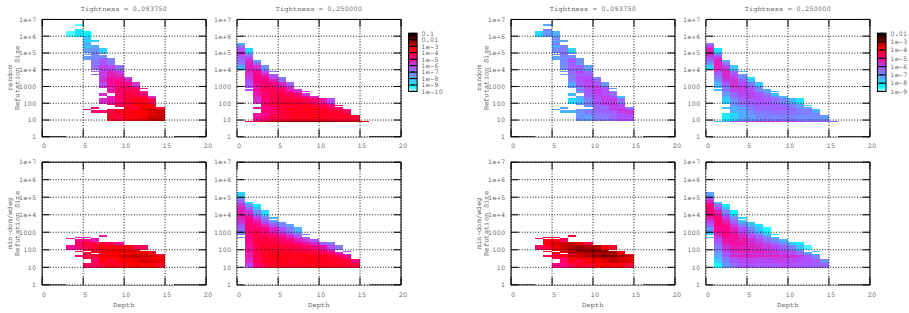
For random problems 17×8 , as well as for QWH-10 with 90% holes, Figure 1 shows heatmaps of the probability of encountering mistakes of a certain size at a certain depth (Figures 1(a) and 1(c)), as well as the proportion of effort spent at a certain depth in refutations of a certain size (Figures 1(b) and 1(d)). Colours represent a log-scale.

Random 17×8 instances, using a random variable ordering with backtrack search exhibit heavy tails in the easy region, but not in the hard region. While a survival function-based analysis [2] can demonstrate the presence or absence of such large mistakes, heatmap visualisations also show precisely the depth where these mistakes occur.

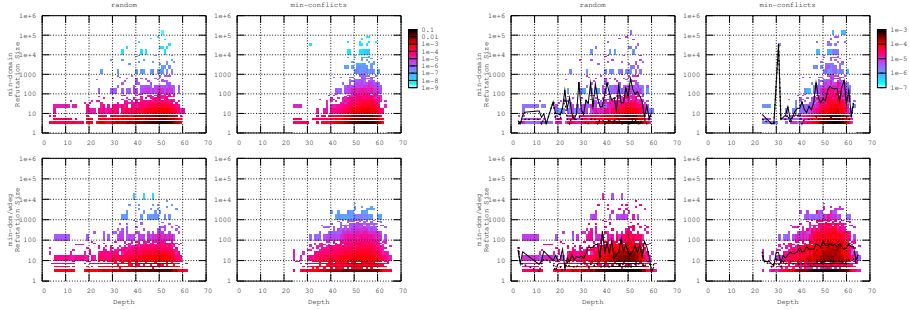
It is also interesting to consider how the size of mistakes varies with depth. The conventional wisdom is that mistakes made at, or near, the top of the tree are exponentially larger than those made deeper in the tree. However, the QWH-10 plots in Figure 1 contradict that assumption – the largest mistakes occur at *intermediate* depths.

The heatmap in Figure 1(d) depicts, *for a population of instances*, the proportion of effort required to refute, for QWH-10 with 90% holes, mistakes of various sizes at each depth. The darkest spots in the heatmap represent those mistake sizes for which the cumulative effort over all the mistakes in our data set was proportionally the largest at that depth. It is clear that over a population of instances, for all the algorithms we used, the bulk of the effort is spent in refuting the extremely large number of small mistakes (4 to 100 nodes) that occur deep down in the search tree (Figure 1(c)).

Random binary problems do exhibit exponential decay of the refutation size with depth (Figures 1(a), 1(b) and 2(d), similar results for random 150×10 and 30×10).



(a) Random 17×8 (left: easy, right: hard): probability of a refutation of a certain size (y-axis) occurring at a given depth (x-axis). (b) Random 17×8 (left: easy, right: hard): proportion of effort spent in refutations of a certain size (y-axis) at a given depth (x-axis).



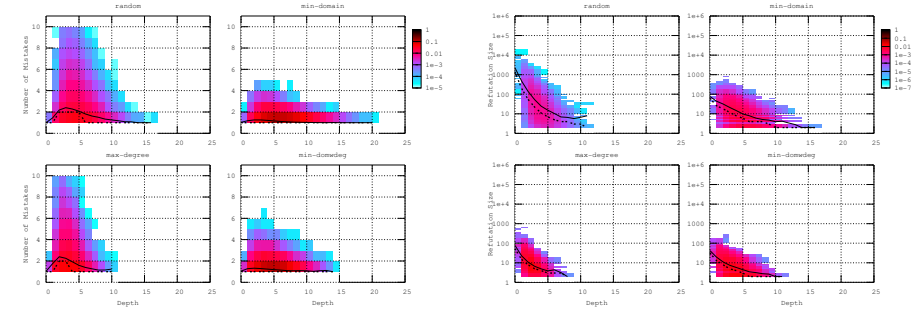
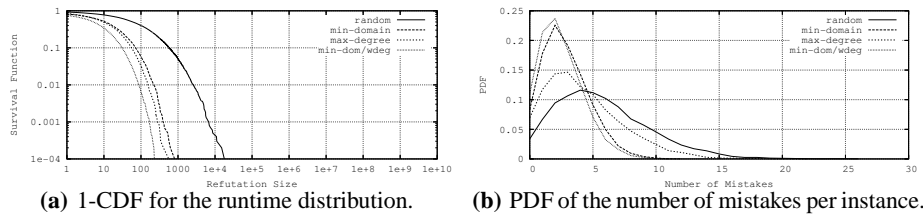
(c) QWH-10 problems with 90% holes: probability of a refutation of a certain size (y-axis) occurring at a given depth (x-axis). (d) QWH-10 problems with 90% holes: proportion of effort spent in refutations of a certain size (y-axis) at a given depth (x-axis). Lines: dotted=medians, continuous=means.

Fig. 1: Heatmaps for random problems 17×8 and QWH-10 with 90% holes.

Moreover, as random problems approach the phase transition, mistakes start occurring close to the root of the tree, and over a population of instances, the bulk of the effort, which corresponds to the dark colour in the heatmaps, shifts towards the top of the tree.

Clearly, the mean and median refutation sizes in Figure 1(d) fail to provide any information with respect to the wide range and distribution of refutation sizes encountered here. The large number of small-to-medium size refutations not only keeps the median below 10, but also prevents the extremely large mistakes from contributing to the mean. Furthermore, the medians and means in Figure 1(d) do not indicate where the effort is in terms of depth. While *for a population of instances* of QWH-10 the effort seems dominated by the disproportionately large number of small mistakes, *for any particular difficult instance*, a small number of large refutations dominate the search effort.

MAC with min-conflicts and min-dom/wdeg is the only algorithm in our arsenal that can eliminate heavy tails for QWH-10 [5]. The bottom-right plot in Figure 1(d) shows



(c) Probability of a certain number of mistakes (y-axis) at a given depth (x-axis) in an instance. Lines: dotted=medians, continuous=means. (d) Proportion of effort spent in refutations of a certain size (y-axis) at a given depth (x-axis). Lines: dotted=medians, continuous=means.

Fig. 2: Sparse random 30×10 : Promise vs fail-firstness; 4 variable orderings, random values.

a far smaller *variation* in the size of the refutations than any of the other 3 plots. This translates into a significant reduction in the variation in instance-based effort and is consistent with the non-heavy-tailed nature of that data set. Similar, but more complex behaviour can be observed in Figure 1(b). The upper-left heatmap is the only one corresponding to a heavy-tailed distribution. There are two characteristics of the other three heatmaps that help in visually determining the absence of heavy-tails: in the lower-left plot there is insufficient variation in the refutation sizes; in the plots on the right, the greatest proportion of our effort is associated with the large refutations.

2.2 Promise versus Fail-Firstness

Good variable ordering heuristics reduce the effort required to refute insoluble subtrees using a property called fail-firstness [4]. Variable ordering heuristics have also been shown to exhibit promise, i.e. they can contribute to a search algorithm’s ability to avoid making mistakes [1]. Promise was measured previously based on the probability that search remains on the path to a solution. Here we measure it very differently and present an alternative analysis of its complex interaction with fail-firstness.

The remainder of our experiments are based on sparse random problems with 30 variables and uniform domain size 10. Figure 2(a) clearly shows that min-dom/wdeg and random variable orderings are the best and worst heuristics, respectively, and that

max-degree performs better than min-domain. Further supporting the usefulness of heatmaps, we can see how Figure 2(b), as well as the means and medians in Figure 2(c), portray min-domain and min-dom/wdeg as being very similar. The heatmaps in Figure 2(c), however, clearly show the mistakes made by the two heuristics are distributed differently across depths, with min-dom/wdeg having a higher probability of making more mistakes per instance over almost the entire range of depths it covers.

Figures 2(c) and 2(d) present a measure of promise and fail-firstness, respectively, for each variable ordering heuristic, arranged left to right and top to bottom in increasing order of performance, as per Figure 2(a). A comparison of the various heuristics depicted there may seem contradictory at first: max-degree seemingly outperforms min-domain due to its better fail-firstness and despite its worse promise (more mistakes), while min-dom/wdeg performs better than max-degree due to its better promise, and despite its slightly worse fail-firstness. Smith and Grant [6] showed that trying harder to fail first does not always improve performance, and our experiments support a similar conclusion for promise. From the heuristics studied here, min-dom/wdeg performs best not because it makes fewer mistakes, or because it refutes them with less effort, but because it strikes a good balance between these two properties.

3 Conclusions

Our novel use of heatmaps nicely complements the use of survival functions and allows a more granular view of the complex interaction between a heuristic's ability to avoid mistakes and its ability to recover from them. Heatmaps have helped show very clearly that the effort required to recover from mistakes is not always correlated with the depth where they occur, and better search heuristics do not necessarily make fewer mistakes, or have the ability to recover from them quickly.

References

1. J.C. Beck, P. Prosser, and R.J. Wallace. Variable ordering heuristics show promise. In *Proceedings of CP-2004*, LNCS 3258, pages 711–715, 2004.
2. C.P. Gomes, C. Fernández, B. Selman, and C. Bessière. Statistical regimes across constrainedness regions. *Constraints*, 10(4):317–337, 2005.
3. C.P. Gomes, B. Selman, N. Crato, and H. Kautz. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *Automated Reasoning*, 24(1/2):67–100, 2000.
4. R.M. Haralick and G.L. Elliott. Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, 14(3):263–313, 1980.
5. T. Hulubei and B. O'Sullivan. The impact of search heuristics on heavy-tailed behaviour. *Constraints*, 11(2–3):157–176, 2006.
6. B.M. Smith and S.A. Grant. Trying harder to fail first. In *Proceedings of ECAI-1998*, volume 14, pages 249–253, 1998.