Heavy-tailed Runtime Distributions: Heuristics, Models and Optimal Refutations

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

Abstract. The question of how to best model a problem in the constraint satisfaction paradigm is receiving considerable attention within the research community. For example, the additional expressiveness and propagation that can be obtained by using global constraints seems to have transformed how modelling is currently approached. In this paper we use a combination of runtime distributions and refutations of failure to enhance our understanding of the quality of a model. We study a continuum of Quasigroup With Holes (QWH-10) models from ones that are entirely binary to ones that are entirely formulated using global constraints. We show that the choice of search heuristics used can impact non-binary models more than it does binary models, but even when there are several global constraints in a model, *inherent* heavy-tailedness can still be observed. While it is not entirely surprising that the use of global constraints does not always monotonically improve the runtime distribution of search, our empirical results indicate that such degradation, when it occurs, is far less noticeable when one looks at the optimal refutations corresponding to the mistakes encountered. However, when global constraints *do* improve the runtime distribution, the refutations encountered are much closer to their corresponding optimal. Last, but not least, we made a remarkable discovery. For the QWH-10 problems studied, all the variable ordering heuristics we used constructed refutations using a very small set of variables that *could* lead to optimal refutations, but did not simply because the variables were selected in the wrong order.

1 Introduction

Constraints researchers have been concerned with the question of what constitutes a good problem formulation for almost two decades [24]. Recent work has focused on the task of heuristically deciding upon a good problem formulation for a search algorithm [5]. The ultimate objective in that work is to automate the process of problem formulation selection.

The realisation that practical problems cannot be adequately modelled using binary constraints has led to a number of developments in the study of the consequences of non-binary formulations [2]. The development of global constraints has significantly changed how problems are modelled in CP [3], since such constraints provide both a rich language for stating the constraints of a problem and, often, ensure that high levels of local consistency can be achieved tractably. Amongst the most widely used global constraints is Regin's ALLDIFFERENT [25], and it has been shown that using this constraint leads to superior formulation of problems [29]. Further developments in modelling have given rise to primal-dual models, in which one uses constraints to channel between alternative formulations in order to benefit from higher levels of inference [8]. The effect of alternative formulations using binary and global constraints, as well as the effects of heuristics, have also been reported [16].

In studying the quality of a given formulation of a problem, or the quality of a given formulation and search heuristic combination, the traditional approach is to analyse runtime statistics gathered from large numbers of runs, such as average/median effort. However, over the past decade we have seen that the use of runtime distributions have given us novel insights into the typical-case complexity of combinatorial search [13] and the effects of problem hardness on search performance [12]. Characterising when runtime distributions exhibit heavy-tails has motivated advances in algorithm development such as rapid random restarts [14, 21].

Summary of Results. In this paper we perform an in-depth empirical study of runtime distributions associated with a continuum of problem formulations for QWH-10 with 90% holes[1] defined between the extreme points where a formulation is entirely specified in terms of binary inequality constraints to models specified entirely in terms of ALLDIFFERENT constraints (global constraints that maintain generalised arc-consistency). For each model we study a variety of variable and value ordering heuristics. Furthermore, we compare the runtime distributions we obtain against runtime distributions where any mistakes made in search are refuted optimally [18]. Through this analysis, we have made a number of key contributions:

1. Firstly, we show empirically that for the problems considered, *variations in the heuristics used have a far more significant effect on hybrid models* (i.e. models using both binary and global constraints) than they do on purely binary models. For example, an improvement in the variable or value ordering heuristic, such as the change from min-domain to brelaz, can cause a dramatic decrease in the search effort (measured in number of nodes) for a hybrid model with 8 global constraints. This is a significant improvement over the binary representation where hardly any difference can be observed between the two aforementioned heuristics.
2. Despite the improvements just mentioned, *runtime distributions of hybrid models can remain inherently heavy-tailed*. While in most cases the algorithms we tried performed better on hybrid models, we show that a straight line can still be observed in a log-log plot of their runtime distributions, even when mistakes are refuted optimally.
3. *Models using global constraints are not always better than purely binary models.* We encountered some configurations in our QWH-10 experiments where we observed that increasing the number of global constraints used to enforce distinct values on rows and columns (and removing the corresponding sets of binary constraints) does not lead to a monotonic decrease in search effort. *The discrepancy all*

[1] We deliberately choose easy problems in order to study heavy-tailed runtime distributions [12].

but disappeared when we looked at the corresponding (quasi-)optimal refutations for the exact same configuration.

4. With the exception of a few unusual cases, the employment of a model that uses global constraints did improve search performance and, when that happened, the *refutations encountered for hybrid models were much closer to their corresponding optimal* than they were for the binary model.
5. Finally, in our quest for reducing the effort required in finding optimal refutations, we hypothesised and verified empirically that, for the problems considered, the variables used in refuting a mistake usually represent only a small fraction of the variables still uninstantiated at the time the mistake was made, *yet this small subset of variables can most of the time be re-ordered to refute the mistake optimally.* Besides the obvious advantage of being able to restrict the search for optimal refutations to that subset and find those optimal refutations faster, this discovery raises a set of other interesting questions.

The remainder of the paper is structured as follows. In Section 2 we give a more precise statement of our motivation. We outline the details of our approach in Section 3, also defining the problem class we study. Section 4 contains a detailed summary of our experiments, with a discussion of the implications in Section 5. We state our conclusions in Section 6.

2 Motivation

Researchers have studied power-law distributions, and heavy-tailed distributions in particular, as part of an extensive effort to better understand and model real-world phenomena, from weather forecasting to stock market analysis. Heavy tails have been used to model the cost of combinatorial search methods and to explain the occurrence of exceptionally hard instances that have been observed amongst certain classes of constraint satisfaction problems, such as such as graph colouring [17], SAT [10], random problems [12, 27, 28], and quasigroup completion problems [13]. In our earlier work we studied the effect of search ordering heuristics on the heavy-tailedness of runtime distributions of binary-formulations of instances of QWH-10 and QWH-20 problems [19].

The characteristics of a search algorithm may depend heavily on the way a problem is modelled [16, 29] – given a set of search algorithms and several alternative ways to formulate a CSP, which algorithm/model combination is the one to choose? For every such combination, one could simply look at the runtime distributions of a statistically significant sample of instances of the problem under consideration. However useful that may be, there are more questions that researchers may wish to answer. If the runtime distribution is heavy-tailed, is it *inherently* so? How close to being optimal are the refutations encountered during search? The same algorithm may have similar runtime distributions for two different models, $\mathcal{M}_1$ and $\mathcal{M}_2$, yet be much closer to optimality when applied to $\mathcal{M}_1$ than when applied to $\mathcal{M}_2$. Having access to the runtime distributions of the optimal refutations [18] would suggest $\mathcal{M}_2$ as the better choice, as there is little that can be done to improve search performance for $\mathcal{M}_1$.

Motivated by these questions, we have set out to study the relationship between heavy tails, ordering heuristics, optimal refutations and global constraints. More specifically, we studied alternative ways of modelling QWH-10 problems with 90% random balanced holes, and a continuum of models ranging from purely binary models to hybrid models and models based entirely on global constraints.

3 The Approach

3.1 Refutations

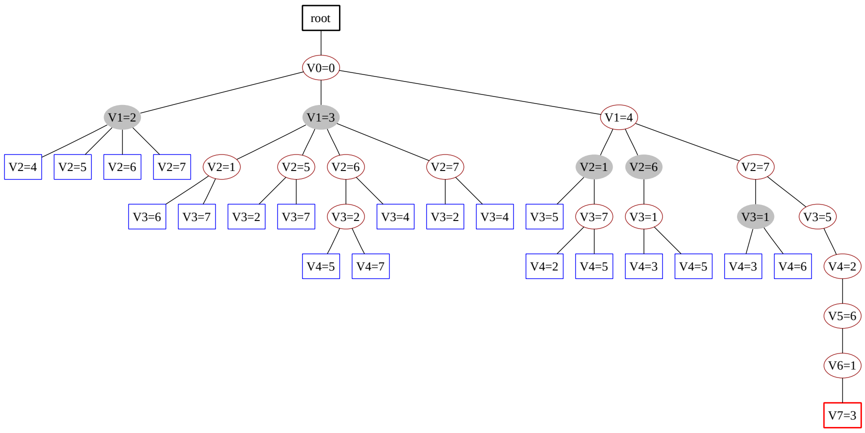
To give a flavour of the kind of analysis reported in this paper, consider the example presented in Figure 1. Figure 1(a) shows MAC's [26] search tree for the 8-queens problem. For simplicity, we used lexical variable and value ordering heuristics, although any other ordering, static or dynamic, would have worked as well. The five grey nodes in the tree are incorrect assignments, or mistakes – there are no paths below them that can lead to a solution, while such paths do exist for their parents. Each node is labelled with its corresponding assignment $V_i = v$, where V_i is an unassigned variable (row on the board) and v a value in its domain (column number). It is easy to see how the subtrees rooted at these mistake points make up the bulk of the search tree. Without them, the only thing the search would spend time on would be restoring arc-consistency after each correct assignment on the path to the solution.

Once a mistake has been made, how quickly can the search recover from it? What is the minimum number of nodes required to prove, for instance, that the second insoluble tree in our example, the one rooted at node "$V_1 = 3$", is indeed insoluble? By changing the order in which variables are selected, we can reduce the number of nodes required to prove insolubility. It can be easily verified[2] that the tree in Figure 1(b) represents an alternative way of proving insolubility for the subtree corresponding to the second mistake. The alternative is made up of only 8 nodes, compared to the 14 in MAC's original (*actual*) refutation. It can also be verified, albeit with more difficulty, that the refutation in Figure 1(b) is optimal in terms of the number of nodes.

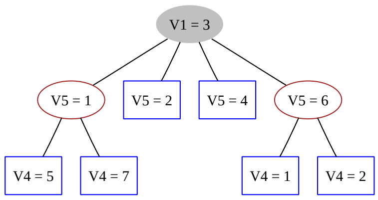
When the optimal refutation is hard to find, we compute the *quasi-optimal refutation*, defined as the smallest refutation whose height does not exceed that of the actual refutation. In our experience it is very rare that the quasi-optimal refutation is larger than the optimal. By accepting quasi-optimality, we can use the height of the actual refutation as an upper bound on the height of the optimal one, dramatically speeding-up the search for better refutations.

Furthermore, as we will see later, it is often the case that our ability to find (quasi-) optimal refutations is not impaired when we restrict the search for a (quasi-)optimal refutation to only those variables involved in the mistake's actual refutation (V_1, V_2, V_3, and V_4 in our example). This can be seen in Figure 1(c), which shows an alternative optimal refutation made up of this restricted set of variables. In general, the search for *restricted (quasi-)optimal refutations* involves only a small percentage of the variables still uninstantiated at the time the mistake was made, and can speed up the search considerably.

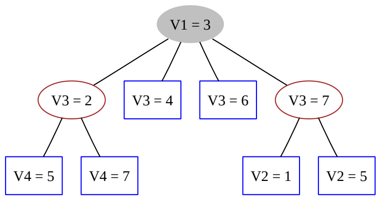
[2] This is left as an exercise for the reader.



(a) MAC's search tree for the 8-queens problem.



(b) Optimal refutation for the subtree rooted at "$V_1 = 3$".



(c) Restricted optimal refutation for the subtree rooted at "$V_1 = 3$".

Fig. 1: MAC's search tree for the 8-queens problem, showing the optimal refutation and the restricted optimal refutation for one of the mistakes made during search.

3.2 Runtime Distributions

In the remainder of the paper we will analyse the runtime distributions of *actual*, *(quasi-)optimal* and *restricted (quasi-)optimal* refutations. Runtime distributions are sometimes characterised by long tails, or *heavy tails*, and are generally modelled using the expression $1 - F(x) = P\{X > x\} \sim Cx^{-\alpha}, x > 0$, where $F(x)$ is the cumulative distribution function (CDF) of a probability distribution function $f(x)$ (PDF), and C and α are constants with $\alpha \in (0, 2)$ and $C > 0$. A near-straight line in a log-log plot for $1 - F(x)$, with a slope equal to $-\alpha$, is a clear sign of heavy-tailed behaviour [13].

3.3 Problem Domain

Our experiments were focused around quasigroup completion problems.

Definition 1 (Quasigroup Completion Problems). *A quasigroup is a set Q with a binary operation $\star : Q \times Q \to Q$, such that for all a and b in Q there exist unique* *elements in Q such that $a \star x = b$ and $y \star a = b$. The cardinality of the set, $n = |Q|$, is called the order of the quasigroup.*

A quasigroup can be viewed as an $n \times n$ multiplication table defining a Latin square, which must be filled with unique integers on each row and column. The Quasigroup Completion Problem (QCP) is the problem of completing a partially filled Latin square. Quasigroup with holes (QWH) are satisfiable instances obtained by starting with a complete Latin square and unassigning a number of cells according to the Markov chain Monte Carlo approach proposed by Jacobson and Matthews in [20]. QWH problem instances are considerably harder when the distribution of the holes is balanced, i.e, when the number of unassigned cells is approximately the same across the different rows and columns [1, 20, 22].

4 Experiments

Our previous studies of heavy-tailed behaviour concentrated on the effects of varying the ordering heuristics [18]. We showed that for certain search algorithms, the runtime distribution is *inherently heavy-tailed*, that is even if we were to use an oracle to refute insoluble subtrees optimally, we would still observe heavy tails. In this paper, we add another dimension to that experiment and study how various ways to model a problem affect the runtime distributions of the algorithms used.

Our experiments were performed on satisfiable QWH-10 problem instances (90 variables) with 90% random balanced holes[3] and included 4 variable ordering heuristics: min-domain [15], min-dom/ddeg[4] [4], brelaz [7] and min-dom/wdeg [6, 23], and 3 value ordering heuristics: random, min-conflicts [9] and its anti-heuristic, max-conflicts, for a total of 12 algorithms. We always broke ties randomly. Most binary instances were too difficult to solve using random variable orderings or variable ordering anti-heuristics, which is why these heuristics have not been included.

With one of the goals of the experiment being the study of various ways to model a CSP, we started by encoding these problems using only binary constraints (propagated using MAC) and then gradually replaced the binary constraints used to enforce distinct cells on rows and columns with equivalent n-ary ALLDIFFERENT constraints (which propagate generalised arc-consistency). For QWH problems, consistency can be enforced on each row and column using either $n(n-1)/2$ binary constraints or one n-ary ALLDIFFERENT constraint, with n the order of the quasigroup. In addition to the binary model, in our experiments we represented QWH-10 using 3 different hybrid models in which we randomly selected 2, 4, and 8 (out of the 20 possible) rows and/or columns and replaced their corresponding binary constraints with a single n-ary ALLDIFFERENT constraint. In the rest of the paper we will use $hybrid = X$ to denote a certain model, with X being 0 for the binary case and 2, 4, or 8 for the others. We

[3] Generated using code based on Carla Gomes' *lsencode* quasigroup generator.

[4] In this paper we abbreviate dynamic-degree as 'ddeg' and weighted-degree as 'wdeg'. The degree of a variable is given by the number of constraints involving the current variable and at least one other uninstantiated variable.

studied settings for X of 16 and 20, but found that these instances were too easy and, therefore, uninteresting.

Using a Beowulf cluster of 96 CPUs over a period of a month we accumulated experimental data on all the 12 variations of MAC, and all the binary and hybrid models described above, totalling over 7.3 million instances, with each individual experiment containing between 20,000 and 100,000 instances. For all variations of MAC and hybrid models, we ran two separate sets of experiments, one in which we computed quasi-optimal refutations and another in which we computed restricted (quasi-)optimal refutations, reporting the cumulative size of each for every instance, alongside the cumulative size of the corresponding actual refutations. We found that in the vast majority of cases[5], the quasi-optimal refutations obtained were in fact *optimal* not only for the binary models [19], but also for the hybrid ones. However, we encountered some instances for which we could only find improved refutations (i.e. refutations for which we were not able to guarantee quasi-optimality), and some for which the search timed out without finding any improved refutation. We included the former in our results, but due to their insignificant number (less than 0.1% of the data [11]), we excluded the latter.

5 Results

Figure 2 and 3 represent half our experiments and show the actual and (quasi-)optimal runtime distributions of our 12 algorithms on the binary and hybrid models. The plots are organised roughly in increasing order of efficiency from left to right and from top to bottom. In the following, we will often extract subsets of these plots to make the interpretation of various points easier. We will use the term *shorter* to refer to refutations that are either optimal, quasi-optimal, or simply the shortest improved refutations we could find that were smaller than the corresponding actual refutations, and the term *restricted shorter* to denote the smallest refutations we could find when the search for optimal refutations was restricted to the variables involved in the actual refutation. Furthermore, we use the term *cumulative effort* to refer to the effort required to refute all the mistakes encountered in a given instance.

As indicated in the introduction, our experiments suggest that more sophisticated models can benefit more from good variable and value ordering heuristics than the equivalent binary models. Figure 4 shows that the performance improvements due to better value ordering heuristics become significant when a good variable ordering heuristic (min-dom/wdeg) is applied to the hybrid=8 model. Similarly, Figure 5 shows that the performance improvements due to better variable ordering heuristics become significant when a good value ordering heuristic (min-conflicts) is applied to the hybrid=8 model. Such differences do not exist for the binary models. While the improvements observed for the hybrid models are closely correlated with the runtime distribution of the (quasi-)optimal refutations, we cannot plot them separately due to lack of space.

Figure 2 shows the runtime distributions of our 12 algorithms on the binary and hybrid models. MAC+min-conflicts+min-dom/wdeg is the only algorithm that succeeds

[5] More than 90% for most experiments, no less than 74% for the others.

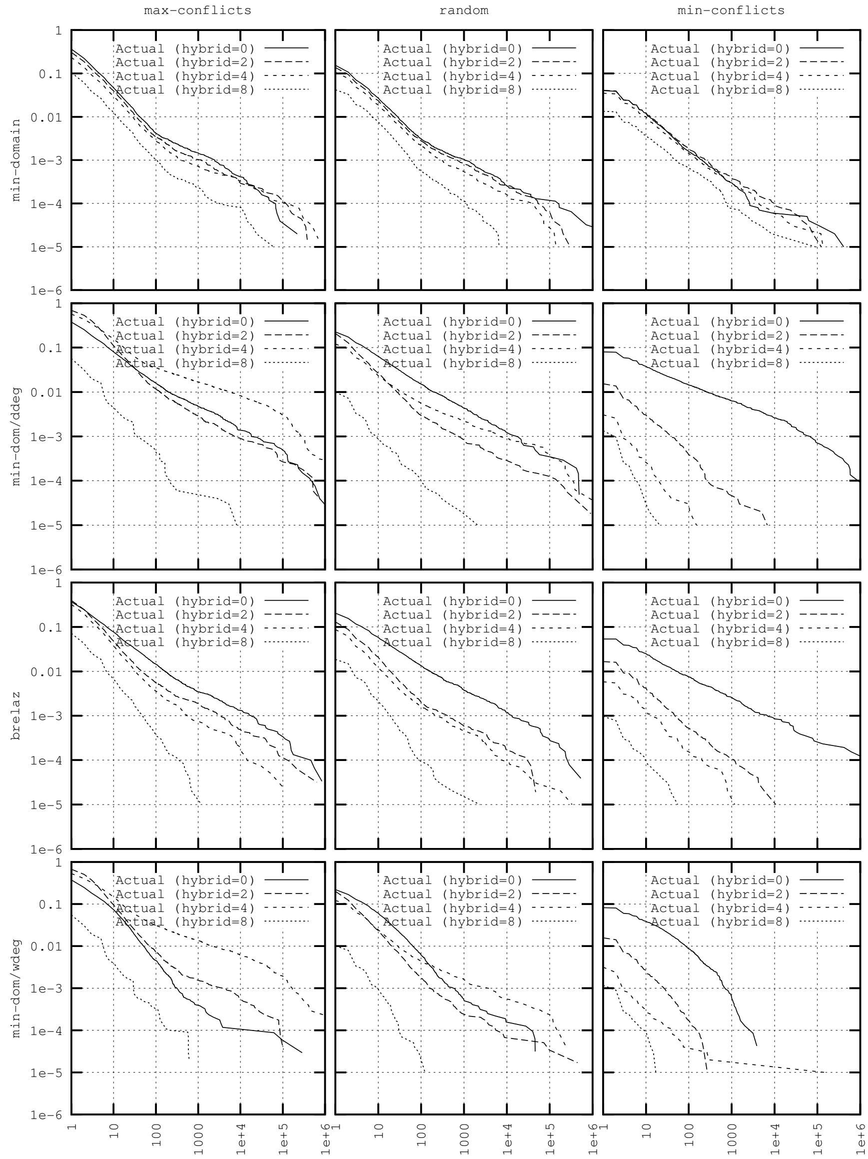


Fig. 2: Complement of the CDF (y-axis) of the cumulative *actual* effort (x-axis). We vary the value ordering across columns and variable ordering across rows.

in eliminating heavy tails for the binary model [19] and continues to do so as we add more global constraints to the model. Interesting enough, all other algorithms remain heavy-tailed for all hybrid models ($hybrid = X$ with $X \leq 8$). Moreover, Figure 3 shows that, for some algorithms, heavy tails do not disappear even when the mistakes encountered are refuted optimally. In other words, for almost all hybrid models studied,

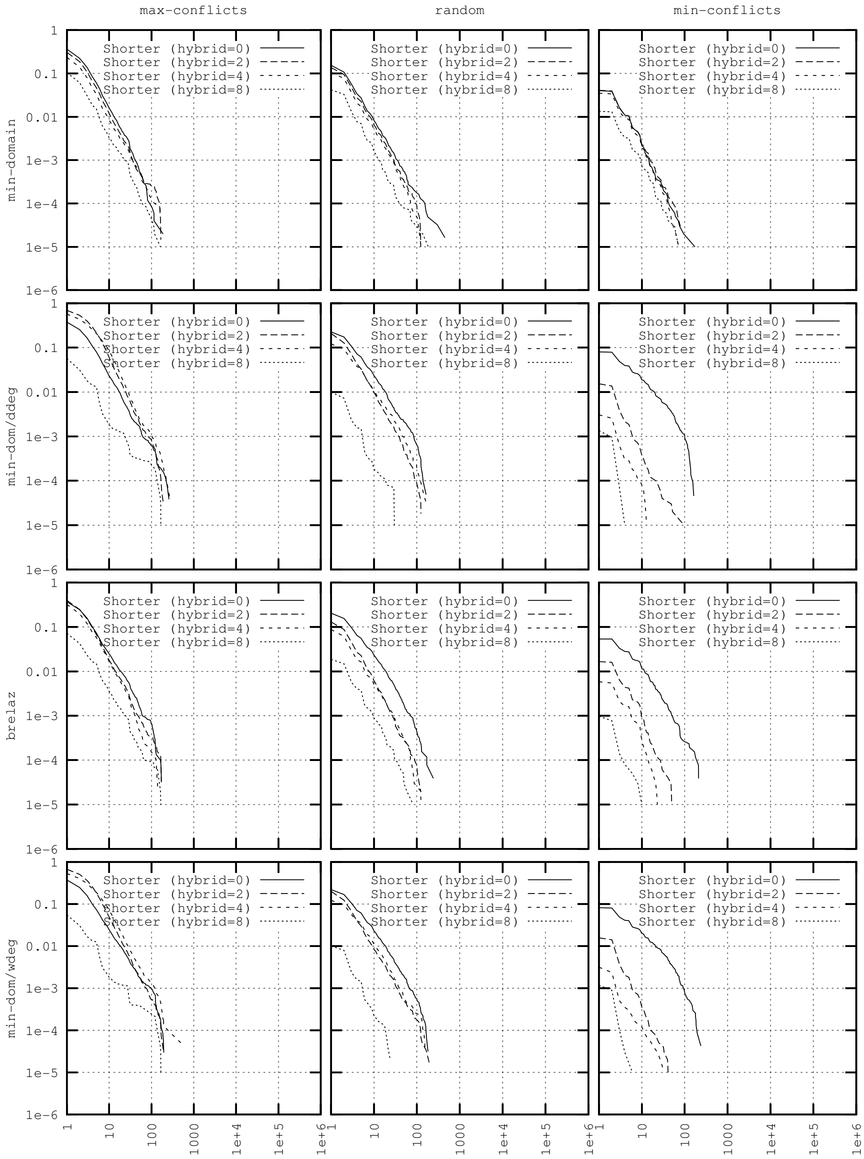


Fig. 3: Complement of the CDF (y-axis) of the cumulative *shorter* effort (x-axis). We vary the value ordering across columns and variable ordering across rows.

even if we were able to use an oracle to refute insoluble subtrees optimally, for some combinations of heuristics we would still see heavy tails (*inherent heavy-tailedness*).

ALLDIFFERENT constraints are known to perform very well when applied to permutation problems such as quasigroups. It is thus quite surprising that despite the addi-

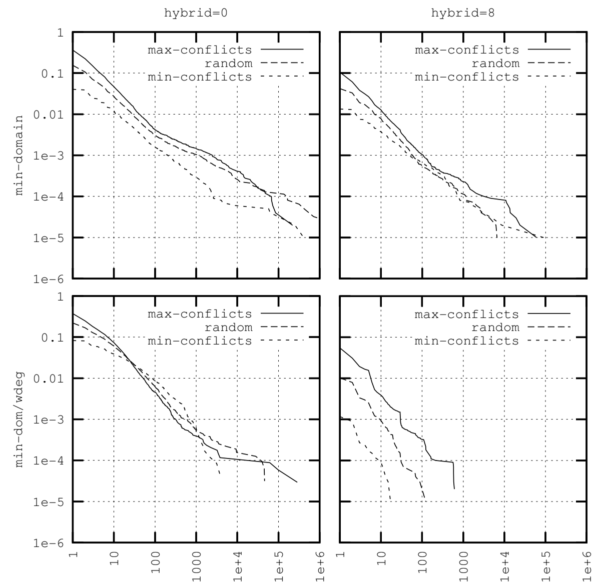


Fig. 4: Complement of the CDF (y-axis) of the cumulative *actual* effort (x-axis) for the binary and hybrid=8 models using a poor (min-domain) and a good (min-dom/wdeg) variable ordering heuristics, combined with various value orderings.

tional propagation they provide, not only do they fail to eliminate heavy tails from the actual refutations, but more importantly, from the optimal ones.

Visually, there are two ways in which heavy tails can disappear from a runtime distribution. One is when the plot is no longer a straight line, something that is usually a reflection of the fact that problems have become uniformly difficult. The other is when the plot continues to be a straight line, but α eventually exceeds 2. This usually happens when problems become too easy [12]. As you can see from Figures 2 and 3, as our models become more sophisticated through the addition of global constraints, the slope of the runtime distributions ($-\alpha$) decreases to the point they can no longer considered heavy-tailed. The way the runtime distributions evolve in our experiments suggests that instances of these problems remain highly irregular as models become more sophisticated, a point that is only strengthen by the similar evolution observed for the corresponding optimal refutations.

Our experiments also show that more sophisticated models do not always lead to better performance. The algorithms that result from combining max-conflicts with min-dom/ddeg and min-dom/wdeg perform worse for the hybrid=4 model than they do for the binary model (hybrid=0), as can be seen in Figure 2. More interesting though is the fact that when we look at the corresponding optimal refutations, while the binary model still outperforms the hybrid=4 model, it only does so by a very small margin.

Figure 6 shows another advantage of models using global constraints, namely the fact that that, compared with binary models, they bring the actual refutations much closer to optimality.

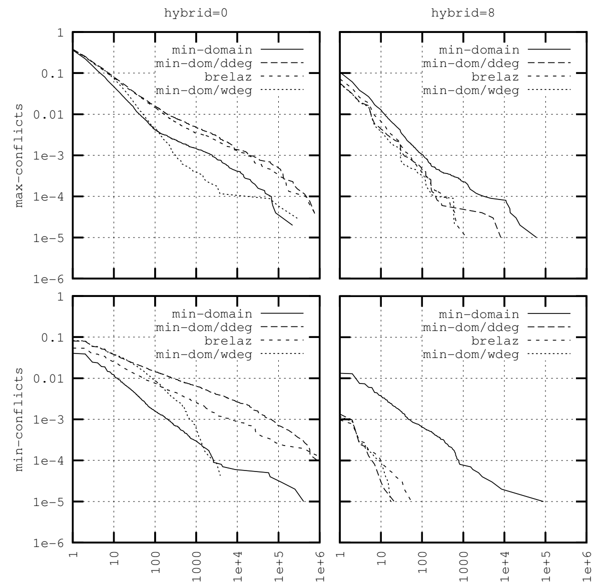


Fig. 5: Complement of the CDF (y-axis) of the cumulative *actual* effort (x-axis) for the binary and hybrid=8 models using a poor (max-conflicts) and a good (min-conflicts) value ordering heuristics, combined with various variable orderings.

Using (quasi-)optimal refutations in analysing models, while new, interesting and helpful, is a process that requires significant resources. As a result, we have spent a considerable amount of time optimising the algorithms computing (quasi-)optimal refutations. One question that seemed interesting from a theoretical point of view and had the potential of significantly speeding up the search for (quasi-)optimal refutations was to restrict the search to only those variables involved in the actual refutation, as per the example in Figure 1.

Remarkably, as Figure 7 shows for the binary and hybrid=8 models, the (quasi-) optimal and the restricted (quasi-)optimal refutations have very similar, most of the time identical, runtime distributions[6]. Even more interesting, for the problems analysed, the average number of variables involved in the actual refutations is only a small fraction of the total number of variables that were still uninstantiated at the time the mistake was made (Figure 8).

These observations suggest that all the variable ordering heuristics studied here have the ability to select a very small subset of the remaining uninstantiated variables, a subset that could be re-ordered to obtain an optimal refutation for each mistake. In terms of proximity to optimality, what differentiates a good heuristic from a poor one is the ability to select those variables *in the proper order*, i.e. an order that would minimise the

[6] Due to time and technical constraints, we ran the *quasi-optimal* experiment and *restricted* experiments in parallel, generating potentially different instances, which explains why occasionally the restricted effort appears to be less than the quasi-optimal.

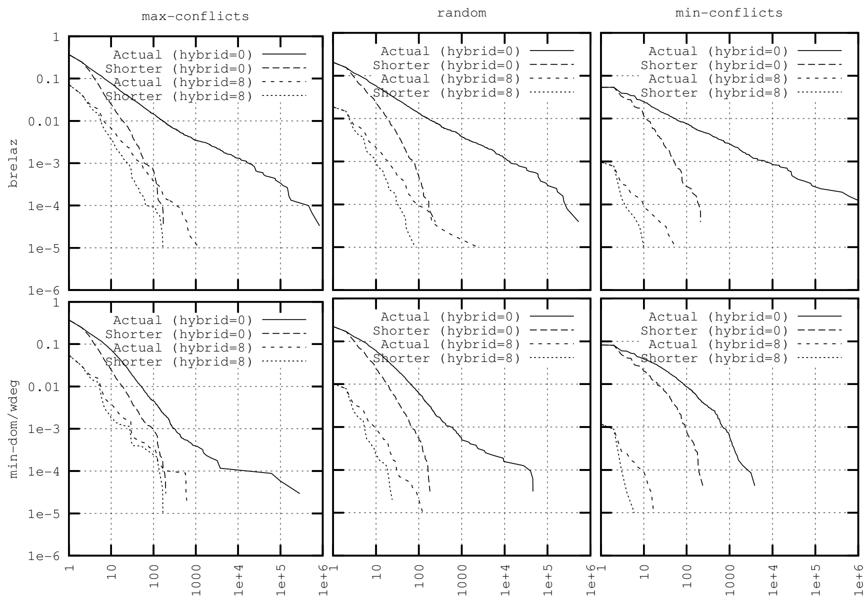


Fig. 6: Comparison of the complement of the CDF (y-axis) for the cumulative *actual* and *shorter* effort (x-axis) for the binary and hybrid=8 models.

size of the resulting refutation. Naturally, a number of interesting questions arise from these observations. Are the variables involved in the actual refutations really special? Would it still be possible to find (quasi-)optimal refutations if we restrict our search to other sets of variables? Do different heuristics select similar sets of variables? We believe our experiments have uncovered a very interesting aspect of search that we intend study further.

6 Conclusions

We have reported on an in-depth empirical study of runtime distributions associated with a continuum of problem formulations of QWH-10, defined between the extreme points where a formulation is entirely specified in terms of binary inequality constraints to models specified entirely in terms of ALLDIFFERENT constraints. For each model we have studied a variety of variable and value ordering heuristics. We have shown empirically that for the problems considered, variations in the heuristics used have a far more significant effect on formulations involving a mix of binary and global constraints than they do on purely binary models. However, the runtime distributions of such hybrid models remain *inherently* heavy-tailed. Models using global constraints are not always better than purely binary models – we encountered some configurations in our QWH-10 experiments where increasing the number of global constraints can degrade search performance. However, based on an analysis of optimal refutations, this is not a fundamental problem with the formulation. Employing global constraints in a model

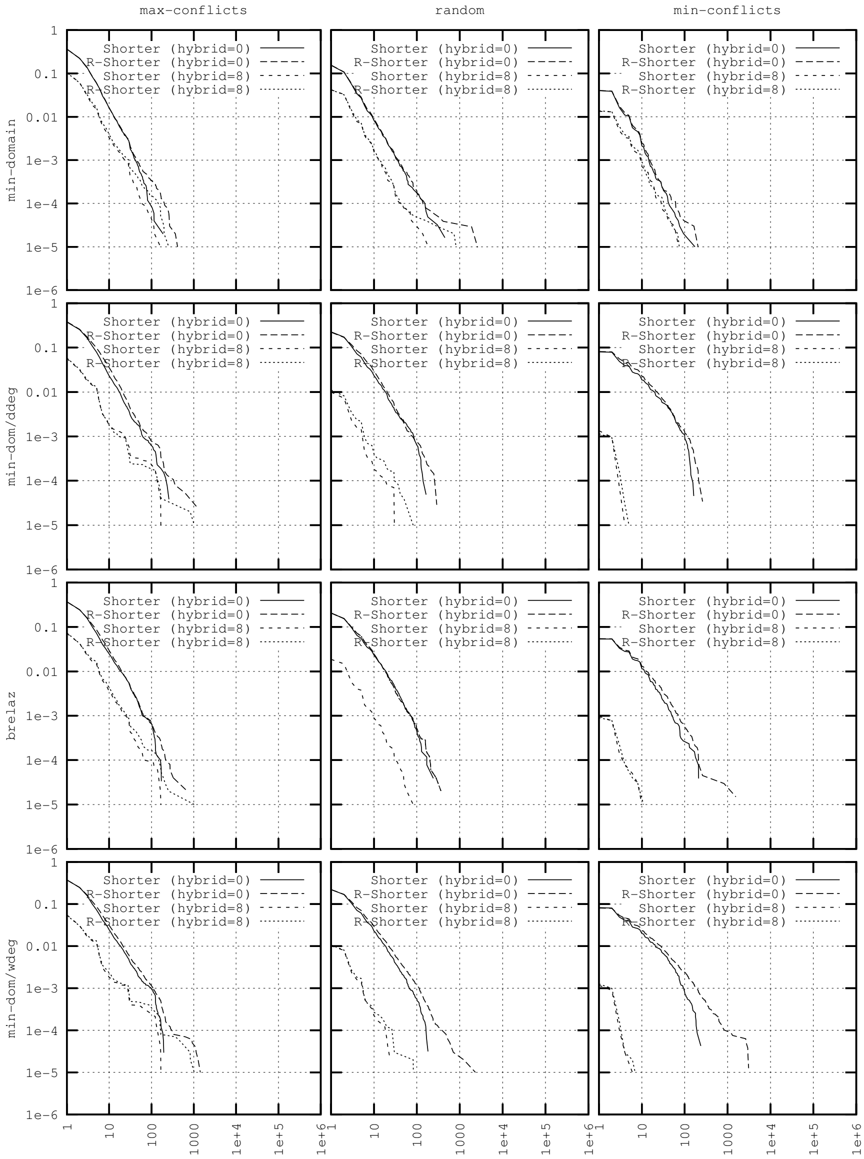


Fig. 7: Comparison of the complement of the CDF (y-axis) for the cumulative *shorter* and *restricted shorter (R-Shorter in the plots)* (x-axis) refutations for the binary and hybrid=8 models.

ensured that search could recover from mistakes with effort closer to optimality. Finally, we report on a very interesting phenomenon, namely that the set of variables that are used by a heuristic to refute a mistake tree during search can be re-ordered to obtain a close to optimal refutation, even though that set is a very small subset of all possible variables that could be used to find the true optimal refutation. This raises a number of

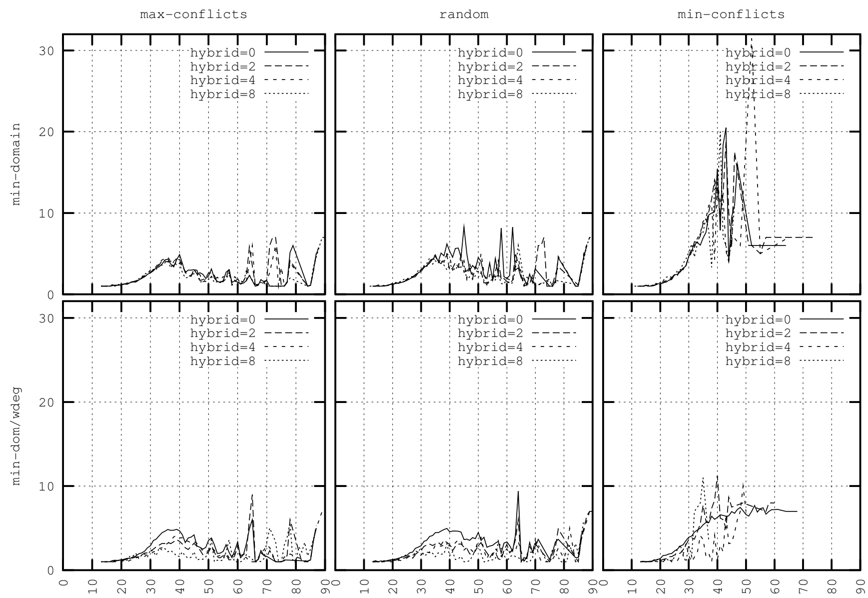


Fig. 8: Average number of variables involved in the *actual* refutations (y-axis) as a function of the number of variables still uninstantiated at the time a mistake was made (x-axis). Medians are very similar. Both binary and hybrid=8 models shown. Remember that the search for restricted (quasi-)optimal refutations is limited to the variables involved in the actual refutations.

interesting questions about why standard heuristics fail to find better refutations, even though they essentially choose the right variables.

Acknowledgements

This material is based on work supported by Science Foundation Ireland under Grant 00/PI.1/C075. We would like to thank John Morrison and the Boole Centre for Research in Informatics for providing access to their Beowulf cluster.

References

[1] D. Achlioptas, C.P. Gomes, H.A. Kautz, and B. Selman. Generating satisfiable problem instances. In *Proceedings of AAAI-2000*, pages 256–261, 2000.

[2] F. Bacchus, X. Chen, P. van Beek, and T. Walsh. Binary vs. non-binary constraints. *Artif. Intell.*, 140(1/2):1–37, 2002.

[3] N. Beldiceanu. Global constraints as graph properties on structured networks of elementary constraints of the same type. Technical Report T2000-01, SICS, January 2000.

[4] C. Bessière and J-C. Regin. MAC and combined heuristics: two reasons to forsake FC (and CBJ?) on hard problems. In *Proceedings of CP-1996*, LNCS 1118, pages 61–75, 1996.

[5] J.E. Borrett and E.P.K. Tsang. A context for constraint satisfaction problem formulation selection. *Constraints*, 6(4):299–327, 2001.

[6] F. Boussemart, F. Hemery, C. Lecoutre, and L. Sais. Boosting systematic search by weighting constraints. In *Proceedings of ECAI-2004*, pages 146–150, 2004.

[7] D. Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.

[8] B.M.W. Cheng, K.M.F. Choi, J.H-M. Lee, and J.C.K. Wu. Increasing constraint propagation by redundant modeling: an experience report. *Constraints*, 4(2):167–192, 1999.

[9] D. Frost and R. Dechter. Look-ahead value ordering for constraint satisfaction problems. In *Proceedings of IJCAI-1995*, pages 572–578, 1995.

[10] I.P. Gent and T. Walsh. Easy problems are sometimes hard. *Artificial Intelligence*, 70(1-2):335–345, 1994.

[11] C.P. Gomes. Randomized backtrack search. *Constraint and Integer Programming*, pages 233–283, 2003.

[12] C.P. Gomes, C. Fernández, B. Selman, and C. Bessière. Statistical regimes across constrainedness regions. *Constraints*, 10(4):317–337, 2005.

[13] 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.

[14] C.P. Gomes, B. Selman, K. McAloon, and C. Tretkoff. Randomization in backtrack search: Exploiting heavy-tailed profiles for solving hard scheduling problems. In *AIPS*, pages 208–213, 1998.

[15] R.M. Haralick and G.L. Elliott. Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, 14(3):263–313, 1980.

[16] B. Hnich, T. Walsh, and B.M. Smith. Dual modelling of permutation and injection problems. *JAIR*, 21:357–391, 2004.

[17] T. Hogg and C.P. Williams. The hardest constraint problems: A double phase transition. *Artificial Intelligence*, 69:359–377, 1994.

[18] T. Hulubei and B. O'Sullivan. Optimal refutations for constraint satisfaction problems. In *Proceedings of IJCAI-2005*, pages 163–168, 2005.

[19] T. Hulubei and B. O'Sullivan. The impact of search heuristics on heavy-tailed behaviour. *Constraints*, 11(2–3):157–176, 2006.

[20] M.T. Jacobson and P. Matthews. Generating uniformly distributed random latin squares. *Combinatorial Design*, 4:405–437, 1996.

[21] H.A. Kautz, E. Horvitz, Y. Ruan, C.P. Gomes, and B. Selman. Dynamic restart policies. In *AAAI/IAAI*, pages 674–681, 2002.

[22] H.A. Kautz, Y. Ruan, D. Achlioptas, C.P. Gomes, B. Selman, and M.E. Stickel. Balance and filtering in structured satisfiable problems. In *Proceedings of IJCAI-2001*, pages 351–358, 2001.

[23] C. Lecoutre, F. Boussemart, and F. Hemery. Backjump-based techniques versus conflict-directed heuristics. In *Proceedings of ICTAI-2004*, pages 549–557, 2004.

[24] B.A. Nadel. Representation selection for constraint satisfaction: A case study using n-queens. *IEEE Expert*, 5(3):16–23, 1990.

[25] J. C. Regin. A filtering algorithm for constraints of difference in CSPs. In *Proceedings of AAAI*, pages 362–367, 1994.

[26] D. Sabin and E.C. Freuder. Contradicting conventional wisdom in constraint satisfaction. In *Proceedings of ECAI-1994*, pages 125–129, 1994.

[27] B.M. Smith. In search of exceptionally difficult constraint satisfaction problems. In *Constraint Processing, Selected Papers*, LNCS 923, pages 139–156, 1995.

[28] B.M. Smith and S. Grant. Sparse constraint graphs and exceptionally hard problems. In *Proceedings of IJCAI-1995*, pages 646–651, 1995.

[29] K. Stergiou and T. Walsh. The difference all-difference makes. In *IJCAI*, pages 414–419. Morgan Kaufmann, 1999.