University of British Columbia
Department of Computer Science
Technical Report TR-99-16

Heavy-Tailed Behaviour in Randomised Systematic Search Algorithms for SAT?

Holger H. Hoos
University of British Columbia
Department of Computer Science

November 30, 1999

Abstract

Prompted by recent results reported by Carla Gomes, Bart Selman, and Henry Kautz, and in the context of my ongoing project with Thomas Stützle on characterising the behaviour of state-of-the-art algorithms for SAT, I measured some run-time distributions for Satz-Rand, the randomised version of the Satz algorithm, when applied to problem instances from various domains. I could not find truly heavy-tailed behaviour (in the sense of the definition used by Carla Gomes et.al.). Instead, I found evidence for multimodal distributions which might be characterisable using mixtures of the generalised exponential distributions introduced in [6]. However, the observed RTDs typically have long tails and random restart, using suitable cutoff times, increases the efficiency of the algorithm, as claimed by Gomes et.al. Furthermore, taking another look at the issue of heavy tails at the left-hand side of run-time distributions, I raise some questions regarding the arguments found in [5].

1 Background

Recently, empirical studies by Gomes et.al. [3, 4, 5] indicated that randomised systematic search algorithms show a behaviour which can be characterised by distributions of the Pareto-Lévy form. More precisely, for hard problem instances from various domains, including SAT, randomised search procedures based on backtracking have shown run-time distributions (cf. [8, 9, 6]) which seem to have an upper tail (or tail on the right-hand side) of the form

$$P\{RT > x\} \sim Cx^{-\alpha}, \qquad x > 0$$

where $0 < \alpha < 2$ and $C > 0$ are constants [5]. When plotting the graph of $G(x) = 1 - P\{RT > x\}$ in a loglog-plot, this heavy tail asymptotically approaches a straight line, reflecting a power-law decay.

This type of behaviour was observed for a variety of algorithms and problem domains, such as Quasigroup Completion, School Timetabling, Logistics Planning, Register Allocation when encoded into SAT or CSP. However, for other SAT instances, such as Random-3-SAT problems, SAT-encoded Blocks World Planning instances, and rarely occuring Quasigroup Completion instances, heavy-tailed distributions could not be observed [4, 5].

Based on this characterisation, it is quite easy to see that random restart will improve the behaviour of these randomised algorithms by forcing the RTD onto an exponential RTD, which (of course) effectively removes the heavy tail (for a formal proof, see [5]). It should be noted, that random restart will not only be effective for heavy tails. Rather, its effectiveness solely depends on the cumulative RTD increasing, at some point, slower than an exponential distribution. Thus, in the presence of any kind of stagnation behaviour, random restart with appropriately chosen cutoff times will improve the probability of finding a solution for run-times greater than a certain limit [6]. Often, knowing the RTD, a cutoff time can be chosen *a posteriori* in such a way that the behaviour of the algorithm is improved for arbitrary or wide ranges of run-times.

From SAT algorithms based on stochastic local search, it is known that under certain conditions, stagnation behaviour does occur and therefore, random restart can be used to effectively improve the algorithm's performance [6, 7]. However, to my best knowledge, heavy-tailed behaviour has not been observed for any SLS algorithm for SAT or other combinatorial problems.

2 Some Empirical Observations

Satz-Rand [4], the randomised version of Satz [13] is one of the best-performing algorithms for solving hard SAT instances known to date. Trying to get a better understanding of the behaviour of Satz-Rand (and other randomised systematic search algorithms for SAT), I measured RTDs for this algorithm when applied to random SAT instances and SAT-encoded problems from various domains, including Blocks World and Logistics Planning [12], and Graph Colouring Problems with a "small world" topology [2]. The Blocks World and Logistics Planning instances have been also used in [4, 5], while the Small World Graph Colouring instances have not been used before in this context.

All experiments reported here use the Satz-Rand implementation obtained from Henry Kautz and available from the SATLIB website. Unless explicitly stated otherwise, a the noise parameter was set to 0.25 — the default value in the Satz-Rand implementation I used (this value seems to work fairly well for most problem instance). Each RTD is based on at least 250 successful runs. The cutoff parameter was always chosen high enough to guarantee that all tries were successful (i.e., the given instance was proven satisfiable of unsatisfiable); typically, a cutoff value of at least $100,000$ backtracks was used. Restart was not used, since I am interested in the performance of the pure algorithm. (The effectivity of random restart can be easily derived from the RTDs of the pure algorithm, as described in [6].)

Blocks World Planning Problems

[4] and [5] observe that for Blocks World Planning instances, Satz-Rand did not show heavy-tailed behaviour. To confirm this, I measured an RTD for Satz-Rand applied to the Blocks World Planning instance bw_large.c. Figure 1 shows two different plots of the same RTD: a standard semilog-plot of the cumulative empirical RTD and a loglog-plot of the associated failure rate $1 - F(t)$, where $F(t)$ is the probability of finding a solution in time $\leq t$. As can be seen from the latter plot, there is no evidence of heavy-tailed RTDs. Interestingly, for noise= 1.0, the upper part of the RTD can be well approximated using a generalised exponential distribution. There is no reason to believe that the behaviour for other Blocks World Planning instances is fundamentally different.

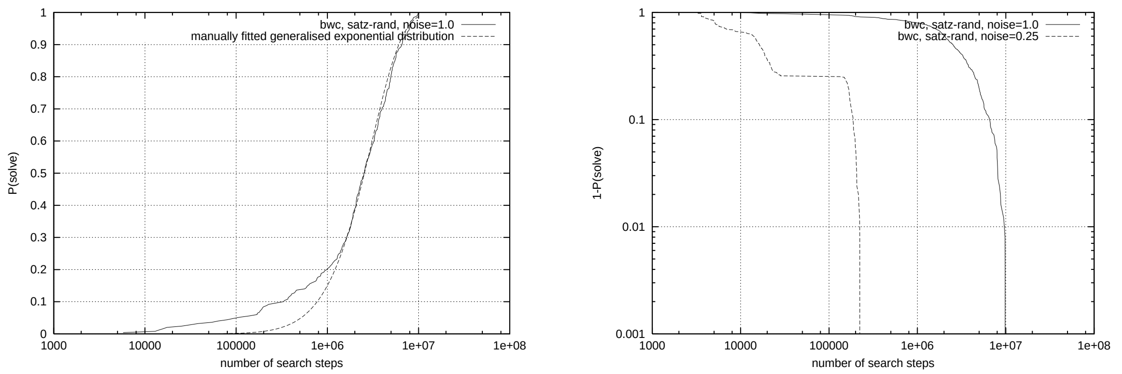


Figure 1: Satz-Rand applied to the Blocks World Planning instance bw_large.c; as can be seen from the right plot, there is no evidence of heavy-tailed RTDs. On the contrary, for noise= 1.0, the upper part of the RTD can be well approximated using a generalised exponential distribution.

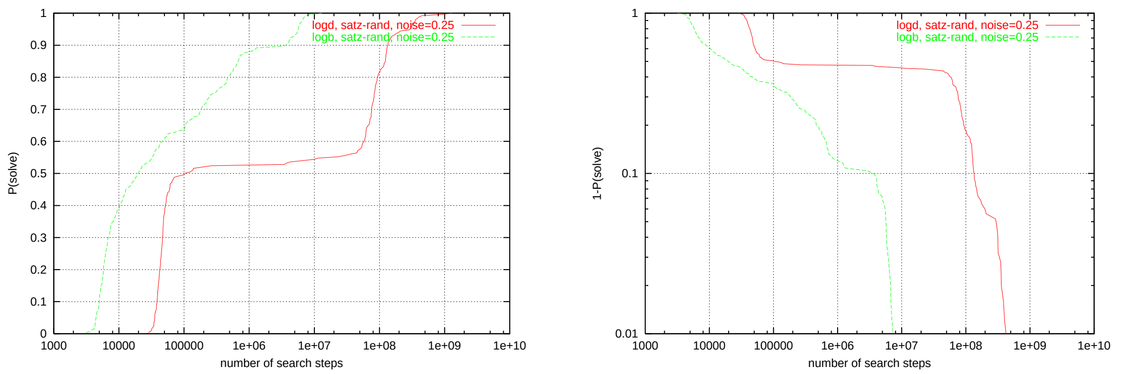


Figure 2: Satz-Rand applied to the Logistics Planning instances logistics.b and logistics.d; as can be seen from the right plot, there is no evidence of heavy-tailed RTDs.

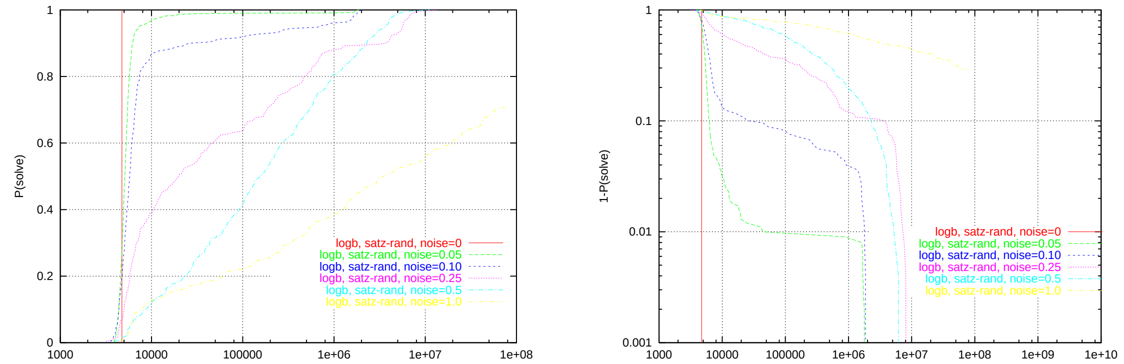


Figure 3: Satz-Rand applied to the Logistics Planning instances logistics.b, using different noise settings. As can be seen from the right plot, for none of the noise settings there is evidence of heavy-tailed behaviour.

Logistics Planning Problems

SAT-encoded Logistics Planning instances are among the problem instances for which heavy-tailed behaviour of Satz-Rand was previously reported [5]. Initially I repeated these experiments only in order to further investigate the RTDs, hoping to finally come up with a functional characterisation. However, as can be seen from Figure 2, I could not find evidence for heavy-tailed behaviour. Instead, there is a very distinctive drop-off in the right part of the failure rate curves for both logistics.b and logistics.d; this corresponds to phases where the algorithm recovers from earlier stagnation behaviour, indicated by flat areas in the cumulative RTD and failure rate graphs. For the larger instance, logistics.d, three stagnation phases can be seen, one of which is very distinctive (over ca. three orders of magnitude in run-time) while the others are less prominent. For logistics.b, the smaller instance, at least two stagnation phases can be detected, one of which is more prominent and occurs around 10^6 steps and another, less significant one just below $100,000$ steps. Note that the areas flanking these stagnation phases correspond to modes of the underlying distributions. Thus the RTDs for both instances are clearly multimodal.

It should be noted that this behaviour is very different from that reported in [5], as can be seen when comparing my Figure 2 with their Figure 9(c). I suspect that they used a different noise setting; however, it should be noted that their failure rate graph covers only a rather small run-time range — I strongly suspect that for higher run-times their failure rate graph will show a behaviour which is qualitatively analogous to the one described here.

It could be possible that heavy-tailed RTDs occur only for specific noise settings. Furthermore, the influence of the noise setting on Satz-Rand's behaviour (as characterised by the RTDs) is interesting in its own right. I therefore measured RTDs for Satz-Rand applied to the smaller Logistics Planning instance logistics.b. As can be seen from Figure 3, for none of the noise settings heavy-tailed RTDs could be observed. When comparing these RTDs, one notices that they become increasingly less steep as the noise is increased. At the same time, the stagnation behaviour reflected by the long tail is more apparent for low noise than for high noise. This agrees with the intuition that higher noise levels reduce the severity of stagnation behaviour. However, as the noise level is increased, the guidance towards finding a solution decreases, which is reflected by the increasingly overall flat shape of the RTDs. For this instance, randomisation (i.e., noise > 0) is not particularly effective, as indicated by the intersections between the RTDs (in particular, see noise=0). However, I would expect that for larger problem instances, such as logistics.d, this situation might be different. Unfortunately, for large problem instances this type of analysis is computationally very expensive, but I expect to obtain the results of an analogous analysis for logistics.d soon.

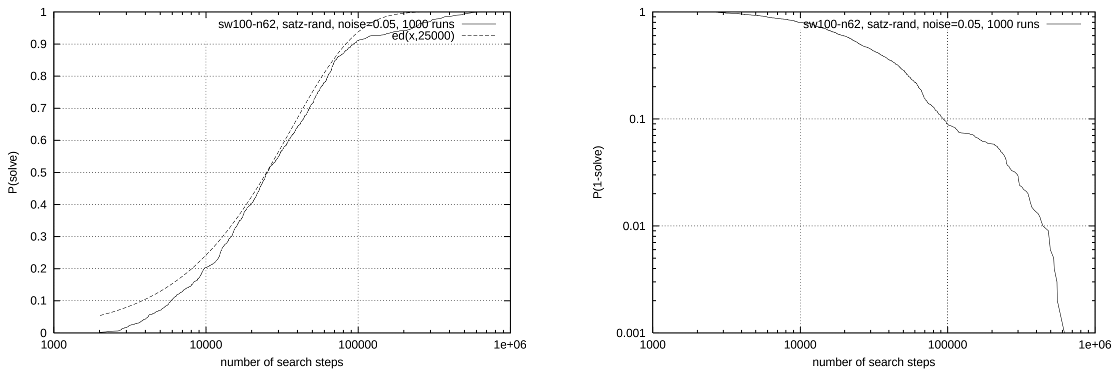


Figure 4: Satz-Rand applied to an exceptionally hard Small Worlds Graph Colouring instance. Again, there is no evidence of heavy-tailed behaviour. As shown in the left plot, the middle of the RTD can be well approximated with a simple exponential distribution.

Small Worlds Graph Colouring Problems

[15] conjectures that for graphs with a so-called small-world property, heavy-tailed behaviour of search algorithms can be observed (see also Section 3.4 of [5]). He gives experimental evidence that the distribution of search cost for a backtrack algorithm *across a test-set of randomly generated problem instances* seems to be heavy-tailed. However, this should not be confused with heavy-tailed *run-time distributions*, which characterise the variation of search cost over different runs of the *same algorithm* applied to *a single problem instance*. Heavy-tailedness for these two types of distributions is not obviously related, and only heavy-tailedness of the latter type (i.e., of RTDs) has direct consequences on the effectivity of random restart when solving individual problem instances. Heavy-tailedness of the former type corresponds to the occurence of exceptionally hard problem instances in instance distributions.

However, there seems to be a possibility that the structure induced by the small-worlds topology also induces heavy-tailed RTDs. To clarify this issue, I measured the RTD for Satz-Rand when applied to sw100-62, an exceptionally hard instance from a random distribution of SAT-encoded Small World Graph Colouring instances [2] with 100 vertices. As can be seen from Figure 4, there is no evidence of heavy-tailed behaviour. Also, again there is clear evidence for stagnation behaviour (around $100,000$ steps) and a multimodal RTD. Note that the plots shown here are based on 1000 runs of the algorithm, therefore they give a rather accurate estimate of the actual RTD and the non-heavy-tailed behaviour reflected by the right part of the graphs (above $100,000$ steps) is to be considered significant.

Interestingly, as shown in the cumulative RTD plot, the middle of the RTD can be well approximated with a simple exponential distribution. It should be noted that of all instances investigated here, sw100-62 is probably the instance which is hardest relative to its problem size. For stochastic local search algorithms, this type of behaviour seems to be typical when applying them to extremely hard problem instances and using good noise parameter settings [6, 10]. The RTDs observed for Satz-Rand on the other problem instances covered here, except for stagnation phases, seem to be steeper than exponential distributions, which again corresponds to SLS behaviour on relatively easy instances. This observation seems to hint at more fundamental aspects of the behaviour of randomised search algorithms; but to clarify this issue, further investigation is definitely required.

Like for logistics.b, I analysed the influence of the noise setting on Satz-Rand's behaviour for this problem instance. As can be seen from Figure 5, there is no evidence for heavy-tailed behaviour. Furthermore, for this problem instance Satz-Rand's behaviour is remarkably robust w.r.t. the noise parameter, as indicated by the very similar RTDs. At this time, I am unsure whether this reflects a property of the problem class or this individual instance.

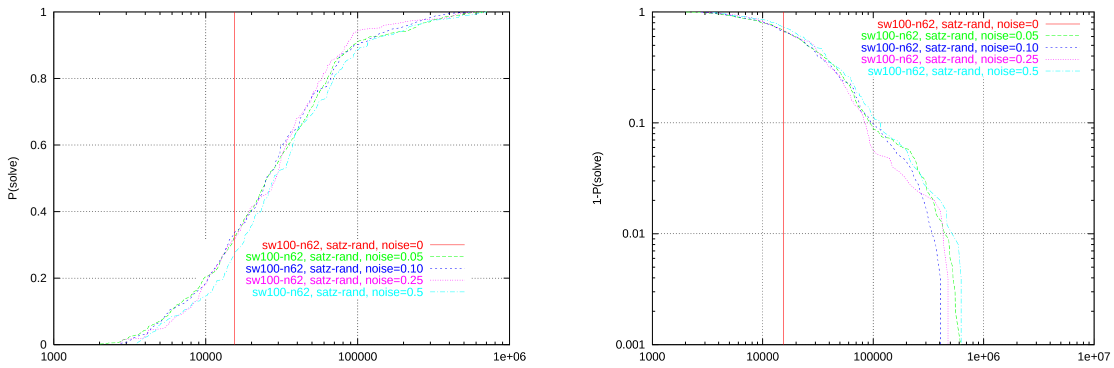


Figure 5: Satz-Rand applied to the exceptionally hard Small Worlds Graph Colouring instance described above, using different noise settings. As can be seen from the right plot, for nonoe of the noise settings there is evidence of heavy-tailed behaviour.

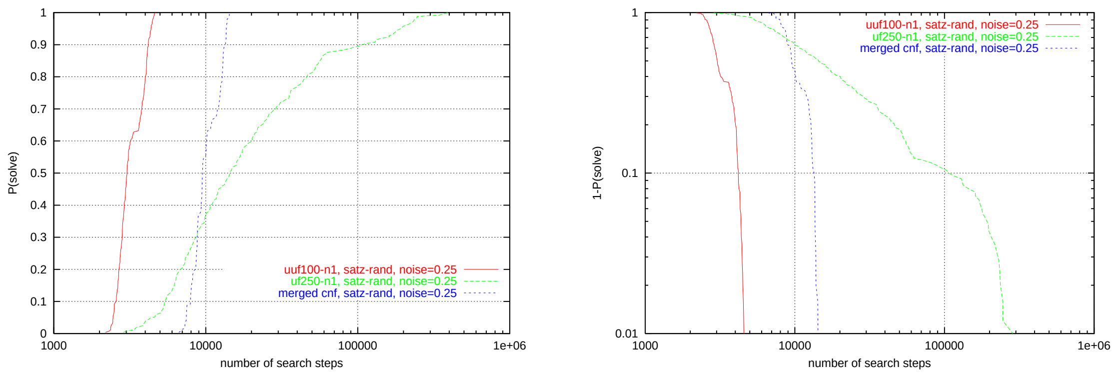


Figure 6: Satz-Rand applied to instance merged-1 obtained by merging a unsatisfiable Random-3-SAT instance with 100 variables and 430 clauses and a satisfiable Random-3-SAT instance with 250 variables and 1065 clauses (no shared variables) and the two component formulae. There is no evidence for truly heavy-tailed behaviour and the behaviour on the unsatisfiable "core" determines the behaviour on the merged formula.

Merged Random-3-SAT Problems

In recent personal communication, David McAllester conjectured that heavy-tailed RTDs might be observed when applying randomised systematic search algorithms for SAT, like Satz-Rand, to instances which are obtained by combining a relatively small, unsatisfiable Random-3-SAT instance and a bigger, satisfiable Random-3-SAT instance. Slightly more generally, I define a class of "merged Random-3-SAT" instances which are constructed by combining two or more individual Random-3-SAT "component" instances such that the variables between the component instances are disjunct (i.e., the components of a merged instance are indepent in the sense that there are no shared variables between components). This definition can be extended to cover weakly or strongly dependent components in a straightforward way.

Note that the question how the solution cost (distribution) of a merged instance depends on the solution cost (distributions) of the component instances is interesting in its own right. For merged instances with independent components, a good SAT algorithm should exploit the structure of the problem. This means that if all components are satisfiable, the solution cost (distribution) should essentially given by the solution cost (distribution) for the hardest component. If there is an unsatisfiable component, the solution cost (distribution) should essentially be given by the

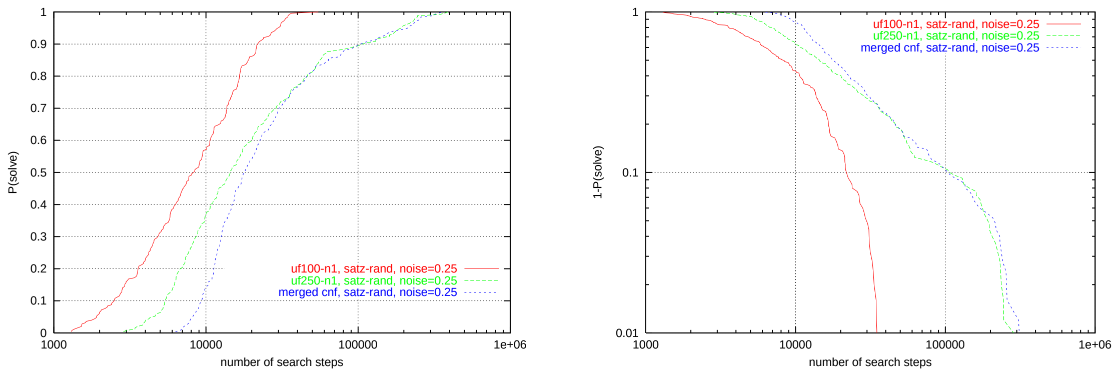


Figure 7: Satz-Rand applied to instance merged-2 obtained by merging a satisfiable Random-3-SAT instance with 100 variables and 430 clauses and a satisfiable Random-3-SAT instance with 250 variables and 1065 clauses (no shared variables) and the two component formulae. There is no evidence for truly heavy-tailed behaviour; the RTDs for the merged formula and the larger component are almost identical for longer runs.

solution cost (distribution) for the hardest unsatisfiable component. Note that in practice, none of the prominent SAT algorithms, both of the stochastic local and systematic search type, explicitly exploits this kind of structure. However, it is imaginable that algorithms like Satz (and Satz-Rand) which are based on sophisticated variable selection heuristics show the desired behaviour nevertheless.

Figure 6 shows the RTD and failure rate graphs for Satz-Rand applied to merged-1, a merged Random-3-SAT instance consisting of one unsatisfiable component with 100 variables and 430 clauses and a satisfiable component with 250 variables and 1065 clauses.[1] Clearly, the shape of the RTDs is very similar for the unsatisfiable component and the merged formula. However, for all percentiles of the RTD, solution cost for the merged formula is approx. 5 times higher than for the unsatisfiable component. It should also be noted that not only the (larger) satisfiable component is uniformly harder to solve, but also the variability in solution cost is much higher than for the unsatisfiable component and the merged instance. Finally, all three RDTs are clearly multimodal and show signs of stagnation behaviour. There is no evidence for any of the three RTDs being heavy-tailed. Overall, Satz-Rand's behaviour on the merged instance is clearly dominated by the unsatisfiable component, although the presence of the independent satisfiable component makes the problem uniformly harder to solve.

Figure 7 shows the results of the same analysis applied to another merged Random-3-SAT instance, merged-2, consisting of two satisfiable components of 100 variables, 430 clauses and 250 variables, 1065 clauses (same as for merged-1). Here, the shape of all three RTDs is similar; the upper parts of the RTD for the merged instance and RTD for the larger (and harder) satisfiable component are almost identical. Different from the merged formula and the larger (dominant) satisfiable component. The RTD for the smaller satisfiable component does not show clear evidence for stagnation behaviour, nor is it obviously multimodal. None of these RTDs appears to be heavy-tailed.

As can be seen from Figure 7, Satz-Rand's RTD on the merged instance merged-2 shows a power-law decay in the failure rate over ca. one order of magnitude in run-time. For yet higher run-times, however, I observe a faster decay which provides evidence against heavy-tailed behaviour. To make sure that this observation is significant and not only causes by the rather small number of samples in this part of the RTD, I measured the same RTD using higher numbers of samples. As can be seen in Figure 8, with increasing number of tries, i.e., an increasing accuracy of the empirical

[1] I did not pick a smaller unsatisfiable component, as Random-3-SAT instances with fewer variables are typically solved by Satz-Rand's preprocessing, i.e., without search.

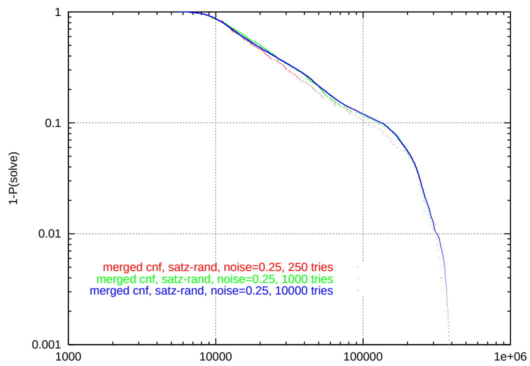


Figure 8: Satz-Rand applied to instance merged-2 from above; the graphs show that the shape of the RTD (which indicates that the upper tail is not heavy) does not change significantly when the sample size is increased.

RTD, the evidence against a heavy-tailed behaviour is confirmed. This situation is typical for all experiments described in this note.

3 Heavy Tails on the Left-hand Side

In Section 3.3, [5] report: "Similarly, in further experiments, we have found that, when dealing with relatively hard problem instances, short runs can occur much more frequently than expected. That is, we have heavy-tails on the left-hand side of the distribution". Although no formal definition is given for the notion of heavy tails on the left-hand side of a distribution, later they make clear that they refer to "a polynomial increase in probability mass over several orders of magnitude", probably having a definition like the following in mind:

> A probability distribution $F(x)$ is heavy-tailed on the left-hand side iff
>
> $$\lim_{x \to 0} F(x)/Cx^{\alpha} = 1, \quad x > 0$$
>
> with constants $C > 0, \alpha > 0$.

This is intuitively analogous to the definition for heavy tails on the right-hand side of a distribution; note that heavy-tailedness on the l.h.s. is reflected by the fact that in a loglog-plot, the RTD graph asymptotically approaches a straight line for $x \to 0$.

Interestingly, heavy tails on the l.h.s. are characteristic of two standard families of distributions which are well-known from fundamental statistics and reliability theory: the exponential and Weibull distributions. As exponential distributions are special cases of Weibull distributions, it is sufficient to show that this claim holds for Weibull distributions. The cumulative distribution function of a Weibull distribution can be given by

$$W(m, \beta, x) = 1 - 2^{-(x/m)^{\beta}}$$

where m is the median of the distribution and β a parameter which controls the coefficient of correlation (standard deviation / mean).

To prove the proposition, we have to show that for all $m > 0, \beta > 0$, there exists $C > 0, \alpha > 0$ such that

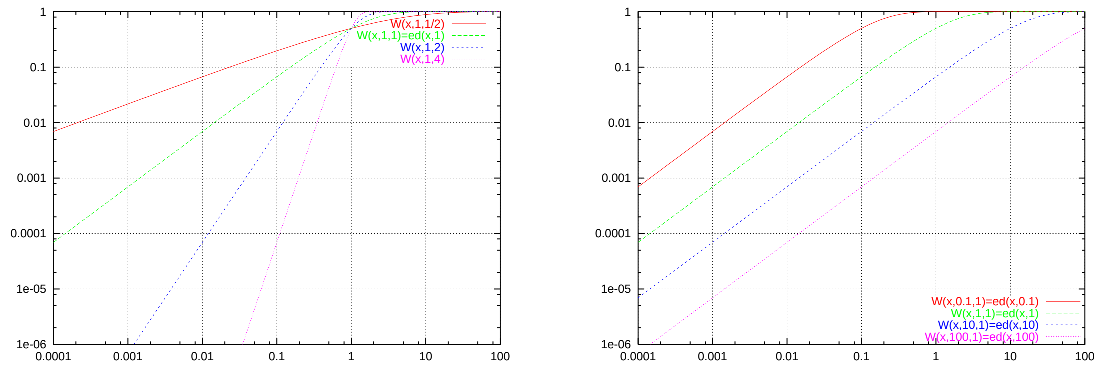


Figure 9: Weibull distributions for different parameter values m, β. Note the heavy-tailed behaviour on the l.h.s.

$$\lim_{x \to 0} W(m, \beta, x)/Cx^{\alpha} = 1. \tag{1}$$

Note that for all m, β there exist m', β' such that $W(m, \beta, x) = 1 - \exp(-(x/m')^{\beta'})$. Using the power series representation

$$exp(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

and letting $y = (x/m')^{\beta'}$, we get

$$\begin{aligned} W(m, \beta, x)/y &= (1 - \sum_{n=0}^{\infty} \frac{(-y)^n}{n!})/y & (2) \\ &= (- \sum_{n=1}^{\infty} \frac{(-y)^n}{n!})/y & (3) \\ &= \sum_{n=1}^{\infty} \frac{(-y)^{n-1}}{n!} & (4) \\ &= 1/1! - y/2! + y^2/3! \pm \cdots & (5) \end{aligned}$$

The latter expression obviously converges towards 1 as $x > 0$ approaches 0, which proves Equation 1 and hence, the original proposition.

Note that this proof not only shows that exponential and Weibull distributions have heavy tails on the l.h.s., but also that each distribution with a heavy tail on the l.h.s. asymptotically approaches exactly one Weibull distribution (with $m' = C^{-1/\alpha}$ and $\beta' = \alpha$). Figure 9 illustrates this result graphically, showing $W(m, \beta, x)$ for various values for m and β.

Based on the argument given before, we know now that different from heavy tails on the r.h.s., those on the l.h.s. are encountered for standard families of distributions. What makes this argument more interesting, is the fact that exponential and Weibull distributions could be successfully used to approximate the behaviour of stochastic local search algorithms. Viewed in this light, the results in [5] seem to suggest that the behaviour of randomised systematic search methods, such as Satz-Rand, might not be too different from that of SLS algorithms as characterised in [6].

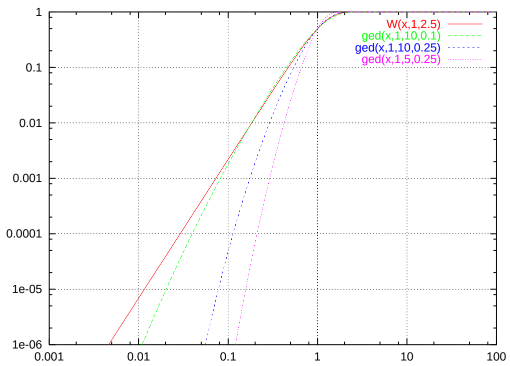


Figure 10: Generalised exponential distributions do not exhibit a true heavy tail on the l.h.s.; this is used to model the behaviour of search algorithms which typically need some time before operating at peak efficiency.

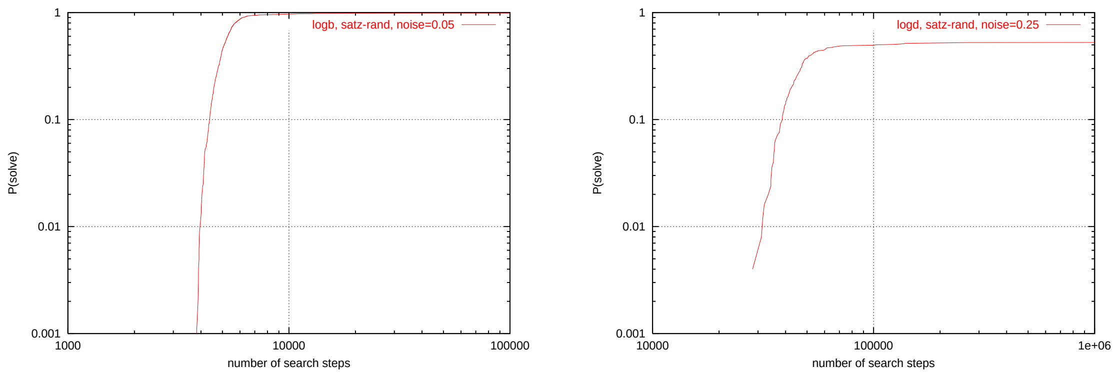


Figure 11: RTDs for Satz-Rand applied to Logistics Planning instances logistics.b (left) and logistics.d (right). Note that heavy tails on the l.h.s. are apparently not present.

However, there is one more wrinkle to this picture. In [6], it was also shown that for very short run-times, the RTDs of SLS algorithms generally tend to fall below the exponential distributions which accurately characterise their behaviour for larger run-times. This was interpreted as an effect of the initial hill-climbing phase of the algorithm and lead to the definition of a new family of *generalised exponential distributions*, which allow to model this observation. However, as can easily be seen from the loglog-plots of the graphs of generalised exponential distributions, these have generally no heavy tail on the l.h.s., although for some parameter values, they may show the same almost-linear behaviour as Weibull and exponential distributions over several orders of magnitude. Figure 10 illustrates this phenomenon.

This raises the question, whether for the randomised systematic search algorithms considered in [5], the RTDs are truly heavy-tailed on the l.h.s., or whether in fact for extremely small run-times the RTDs will also "fall off" the straight line in a loglog-plot, i.e., exhibit similar behaviour as previously observed for SLS algorithms [6]. Indeed, revisiting their results, at least one of the two graphs they show to support their finding of heavy-tails on the l.h.s. indicates that this could be the case (see Figure 9(a) in [5]). The data for the Logistics Planning instance shown in their Figure 9(b), which seems to be more relevant in this context, as it shows the behaviour of a SAT algorithm while Fig. 9(a) refers to a CSP algorithm, does not indicate this phenomenon. However, this RTD data is extremely sparse (only 9 data points) and it seems to be possible that a better empirical estimate of the true RTD would give a result similar to the one in their Fig. 9(a).

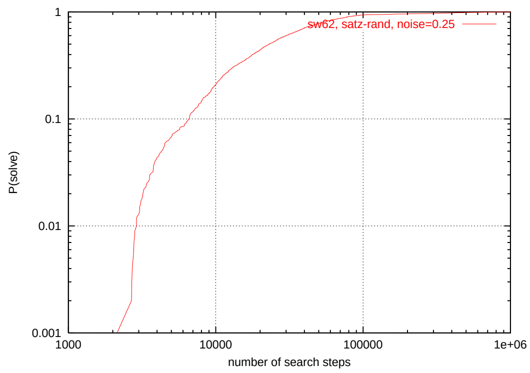


Figure 12: RTDs for Satz-Rand applied to the extremely hard Small World Graph Colouring instance sw100-62. There is no evidence for a heavy tail on the l.h.s.

To shed some more light on this issue, in Figures 11 and 12, I show loglog-plots of the RTDs for Satz-Rand when applied to the two Logistics Planning instances, logistics.b and logistics.d, and the extremely hard Small Worlds Graph Colouring instance sw100-62, described above (noise = 0.25, 1000 tries each, all successfull). As can be clearly seen from these RTD graphs, the tails on the l.h.s. are not truly heavy but clearly "fall off' the straight lines which approximate the RTDs graphs for slightly larger run-times.

These empirical results seem to indicate that the RTDs for randomised systematic search methods are not heavy-tailed on the l.h.s.; rather for very short run-times, these algorithms seem to show a behaviour similar to that of SLS algorithms as described in [6]. Intuitively, this observation indicates that randomised systematic search procedures also need an "initialisation phase" before they reach their full effectivity. This is consistent with an observation found in [5], Footnote 12 (page 24) that "at least a small number of backtracks is required [to effectively solve hard instances]".

4 Approximating Satz-Rand's RTDs with GED Mixtures

For various reasons (see [6, 10, 11]) it would be desirable to functionally approximate the RTDs for Satz-Rand. Obviously, due to the fact that these distributions seem to be typically multimodal, simple classes of distribution functions known from statistical literature are not suitable here. Due to the fact that for some of the instances analysed above, there seemed to be sections of Satz-Rand's RTD which could be approximated by exponential or generalised exponential distributions [6, 11], I decided to try an approximation using mixtures of generalised exponential distributions.

In [6], I introduced the family of generalised exponential distributions to model the behaviour of SLS algorithms. The cdf of a generalised exponential distribution (ged) is given by:

$$ged[m, \gamma, \delta](x) = 1 - 2^{-(x/m)^{1+(\gamma/x)^{\delta}}}$$

Although the full defining term looks quite complicated, the definition simply reflects the idea of a Weibull distribution with a dynamically changing β parameter. More precisely, the new distribution is obtained from a Weibull distribution by introducing a hyperbolically decaying β parameter. Like for the exponential and Weibull distributions, m is the median of the distribution. For SLS algorithms, the two remaining parameters intuitively correspond to the length of the initial search phase (γ) and to its impact on the overall behaviour (δ). High γ values indicate a long initial search phase, while high δ values are used to model a strong influence of the initial search phase.

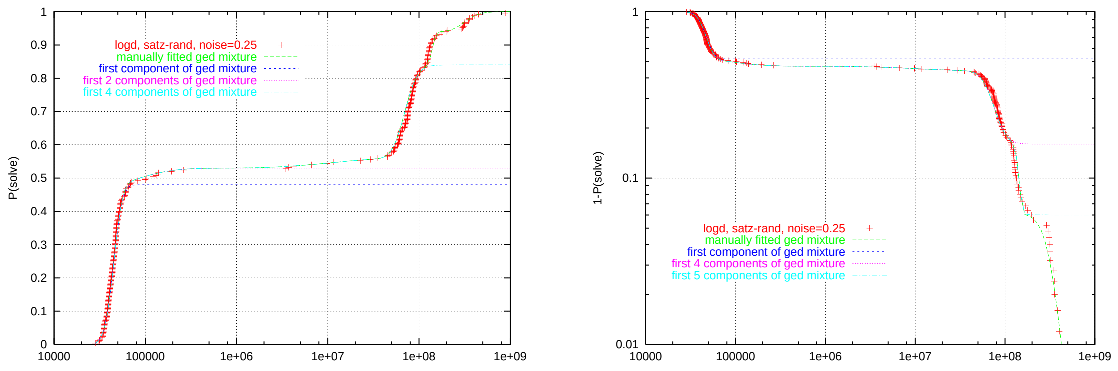


Figure 13: RTDs for Satz-Rand applied to logistics.d and approximation with ged mixture model.

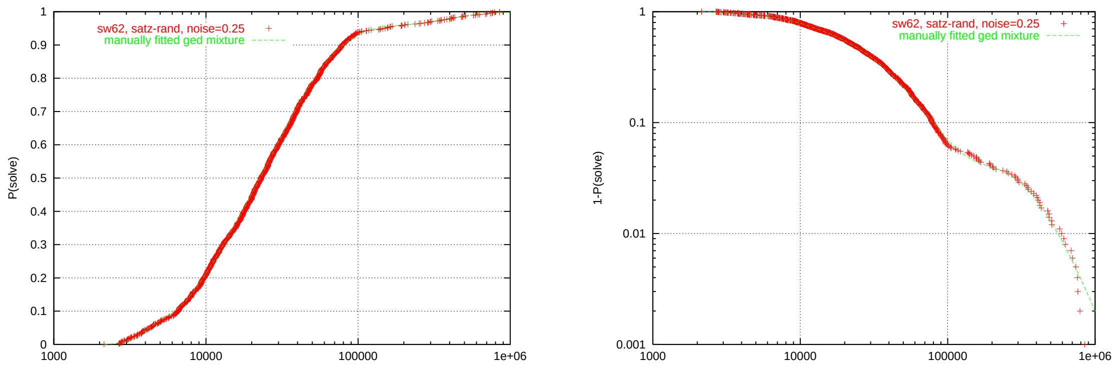


Figure 14: RTDs for Satz-Rand applied to sw100-62 and approximation with ged mixture model.

Note that the exponential distribution is a special case of this new class of distributions, since $ed[m] = ged[m, 0, \delta]$, while generally, Weibull distributions cannot be represented within this family. As can be easily seen from the definition, $ged[m, \gamma, \delta](x)$ asymptotically approaches an exponential distribution $ed[m](x)$ for large x.

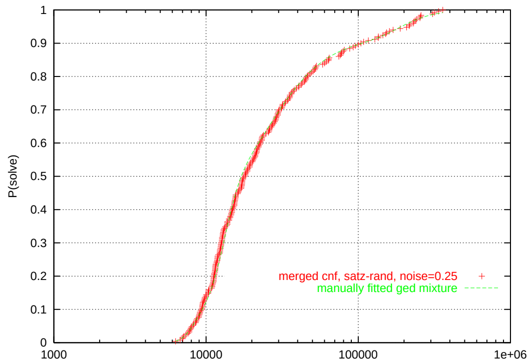
Mixtures of these distributions are characterised by the folloging cdf:

$$\sum_{i=1}^{\nu} c_i \cdot ged[m_i, \gamma_i, \delta_i](x)$$

For SLS algorithms, the motivation behind using these mixtures for modelling their RTDs is based on the idea that dependent on the actual search position, different parts of the search space dominate the algorithm's behaviour. As long as the search space is sufficiently homogeneous, this is not relevant and a simple ged suffices to model the observed RTDs. If however, the search space contains drastically different regions and the algorithm is sensitive to these differences, mixture models are needed. An analogous argument should be applicable to the randomised systematic search methods studied here.

Fitting observed data with these mixture models is quite difficult, and I have not found a good way of automating this procedure. Therefore, in the following I give some anecdotal evidence for the goodness of the RTD approximations using ged mixtures, where the model was manually fitted to the data. It should be noted that for none of the RTDs I tried this, the manual fitting proved to be extremely difficult or impossible.

Figures 13 to 15 show the approximations of Satz-Rand's RTDs using the ged mixture model. As



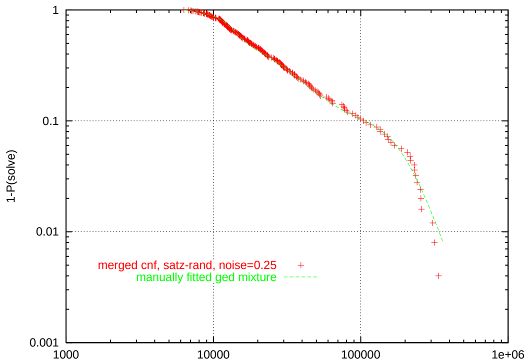


Figure 15: RTDs for Satz-Rand applied to merged-2 and approximation with ged mixture model.

can be seen from the graphs, the approximations are very good. None of the models use more than six component geds, and in all cases reasonable fits can be achieved with a smaller number of components. It should also be noted that most probably the noise settings used here are not optimal. It is possible that for optimal noise levels the RTDs can be approximated by simpler mixture models. While the preliminary results presented here suggest that ged mixtures seem to be a reasonable model for Satz-Rand's behaviour over a range of instances, to answer the question whether this characterisation is as universal as the analogous one for SLS algorithms [6], further investigation is required. At this time, my hypothesis is that the behaviour of Satz-Rand can be generally well approximated with ged mixtures for reasonably good noise settings. Furthermore, I would not be surprised if this characterisation would generalise to other randomised systematic search algorithms such as RELSAT-Rand.

5 Discussion

Although it would be conceptually very nice to characterise the behaviour of search algorithms for hard problems using non-standard, heavy-tailed distributions known from theoretical statistics and real-world phenomena, I was not able to get solid evidence for such a characterisation when analysing the run-time behaviour of Satz-Rand applied to hard SAT instances from various domains, including some of those used in [5]. Rather, the experimental results reported here seem to indicate that there is no truly heavy-tailed behaviour on either side of the RTDs.

There are instances, in which small samples of the RTDs and rather low cutoffs might suggest the existence of heavy-tails on either side, as indeed approximately polynomial decay of the tails (i.e., close to linear sections of the RTD in appropriate loglog-plots of the tails) over one or more orders of magnitude can be observed for some instances. However, better estimates of the actual RTDs indicate clearly that for the cases (instances and noise values) considered here, this behaviour seems not to be present for higher (resp. lower) run-times, which means that true heavy-tailed behaviour does not occur.

These results are not entirely conclusive, as the dependence of the RTDs on the noise parameter, the algorithm's behaviour for other problem instances (such as the ones from [5] not covered here), and finally other algorithms (Relsat-Rand, randomised systematic CSP algorithms) remains to be further investigated. However, the results presented here seem to cast some doubts on the universality of heavy-tailed RTDs for randomised systematic search algorithms applied to hard combinatorial problems.

At the same time, I suggest an alternate mathematical model for characterising these algorithm's behaviour which is not only more consistent with the data presented here, but also covers the behaviour of stochastic local search (SLS) algorithms for SAT [6]. In some sense, this model is theoretically less elegant, as it is based on mixtures of distributions and is, at this time, difficult to automatically fit to the data. However, at least for SLS algorithms, this model has some plausible justification based on the intuitive behaviour of the respective algorithms.

While, if confirmed by further experiments, this could lead to a fundamentally different characterisation of search behaviour than the one given in [5], another contribution of [5], namely the effectivity of randomisation and random restart for improving the performance of systematic search algorithms like Satz-Rand (cf. their Section 4), remains mostly unaffected. As indicated in [5] and argued here, the effectivity of random restart does not rely on heavy-tailed behaviour, but solely on the presence of stagnation behaviour (in the sense discussed above).

However, to clarify the general value of randomisation and restart techniques, it seems to be necessary to address the following issues in more depth:

1. How frequently doe stagnation behaviour occur for randomised systematic search algorithms?
2. How can stagnation behaviour be effectively detected while the algorithm is running?
3. How robust is the performance of randomised search algorithms w.r.t. the setting of the noise parameter?
4. Can we find good methods for automatically tuning the noise parameter and cutoff time such that the randomised algorithm shows consistently good performance?

In the following, I discuss each of these issues briefly.

Occurrence of Stagnation Behaviour. While at the time being, there seems to be some good evidence that stagnation behaviour does occur for certain randomised systematic search algorithms, it is not clear whether different, maybe stronger types of randomisation might not eliminate this problem. A (weak) argument for this is the stagnation behaviour observed in SLS algorithms for SAT when the randomisation is not strong enough [6, 7]. However, it might well be that stagnation behaviour cannot be generally avoided (whithout sacrificing competitive performance). In this case, it seems to be useful to get a better understanding on the factors which influence the occurrence and severity of stagnation behaviour, to which end an improved mathematical model of the observed run-time behaviour seems to be advantageous.

Detecting Stagnation Behaviour. The problem of detecting stagnation behaviour is equivalent to the problem of finding good cutoff times when using random restart. *A posteriori*, optimal cutoffs can be easily derived from the RTD data; but as long as we do not have good models which allow us to essentially predict the RTDs of a given algorithm for whole classes of problem instances, this does not help with effectively solving unknown problem instances. One mechanism which has been used in the Operations Research literature for detecting and overcoming stagnation behaviour of SLS algorithms is adaptive or soft restart (see, e.g., [14]). The idea is to keep track of the relative progress of search (measured by the improvement of the objective value function over some fixed period of time) and to restart when this relative progress is not good enough. For systematic search algorithms for SAT which use forward propagation techniques one could try to do something similar by measuring progress in terms of the amount of unit propagations over some number of decision points. Another idea is to base the progress measure on the distribution of scores when selecting a variable to be assigned. Flat distributions of these scores could potentially indicate search stagnation. While generally, it is to be expected that for each adaptive restart mechanism instances can be constructed which render this mechanism ineffective (by "misleading" it into bad decisions), I tend to believe that for most instances, including application-relevant ones, effective adaptive restart strategies can be found.

Robustness of Performance w.r.t. Noise Setting. While the issue of robustness w.r.t. noise settings is apparently not addressed in the literature on randomised systematic search algorithms for combinatorial problems, I believe it is a crucial one. The reason for this is the fact that by introducing a noise parameter to systematic search, even if for some settings the performance is considerably improved, as long as no method is known for automatically finding good noise values for given problem instances during the run of the algorithm,[2] the real interest is in the expected performance — which is obviously affected by the robustness of the algorithm w.r.t. noise settings. Recent results regarding the analogous question for SLS algorithms [11] indicate that for these, a qualitatively different behaviour can be observed when the noise setting is higher than a certain critical value. This critical value, however, varies with problem size and domain; furthemore, with increasing size and hardness of the problem instances, SLS algorithms apparently tend to become more sensitive w.r.t. the noise setting. It would be interesting to see whether a similar situation can be found for randomised systematic search algorithms. Preliminary experimentation seems to indicate that this could be the case. This is intuitively not surprising, as with increasing problem size the effects of "bad" decisions in the search process can be expected to take longer to become visible to the algorithm. At the same time, it seems plausible to imagine that with increasing problem size there is more and more potential to encounter a higher number of these "traps". In any case, the experimental methodology for further investigating this issue is mostly the same as used for SLS algorithm in [11].

Automatic and Robust Tuning of Parameters. This issue has been discussed above to some extent w.r.t. the cutoff time. Regarding the automatic adjustement of the noise parameter it seems to be desirable to first get a better picture of the dependence of the RTDs on the noise. This also allows to address the dependencies between noise setting and cutoff time. It would be very interesting to see whether the fundamental result for SLS algorithms shown in [6, 11], namely the fact that for sufficiently high noise resp. strong randomisation not only the best performance is achieved, but also random restart is ineffective, carries over to randomised systematic search algorithms. Based on preliminary experiments I would expect that for the present randomisation mechanism [4, 5] this is not the case. However, this issue needs to be further investigated, to which end the same methodology as in [6, 10, 11] can be used.

As this discussion indicates, there are many questions to be answered in the context of randomised systematic search methods; at least for some of these the way of finding answers seems to be rather straightforward. Hopefully, further investigation along these lines will shed some light on the issues discussed in this report and lead to an improved understanding and applicability of this promising class of algorithms.

References

[1] Frost, D.; Rish, I.; Vila, L. Summarizing CSP Hardness with Continuous Probability Distributions. *Proceedings of AAAI'97*, pages 327–333, 1997.

[2] Gent, I. P.; Hoos, H. H.; Prosser, P.; Walsh, T. Morphing: combining structure and randomness. *Proceedings of AAAI-99*, pages 654–660, MIT Press, 1999.

[3] Gomes, C.; Selman, B.; Crato, N. Heavy-Tailed Distrbutions for Backtrack Search. *Proc. Third Intl. Conf. on Principles and Practice of Constraint Programming (CP-97)*, pages 121–135, 1997.

[4] Gomes, C.; Selman, B.; Kautz, H. Boosting Combinatorial Search through Randomization. *Proc. AAAI-98*, Madison, WI, 1998.

[2]Again, a posteriori these can be easily determined from RTD data.

[5] Gomes, C.; Selman, B.; Crato, N., and Kautz, H. Heavy-Tailed Phenomena in Satisfiability and Constraint Satisfaction Problems. *To appear: J. Automated Reasoning, Special Issue "SAT'2000"*, 2000.

[6] Hoos, H. H. *Stochastic Local Search — Methods, Models, Applications.* Ph.D. Dissertation, Fachbereich Informatik, Technische Universität Darmstadt, Germany, 1999. Available at `www.cs.ubc.ca/spider/hoos/phd-thesis.html`

[7] Hoos, H. H. On the Run-time Behaviour of Stochastic Local Search Algorithms for SAT. *Proceedings of AAAI'99*, pages 661–666, MIT Press, 1999.

[8] Hoos, H. H; Stützle, T. Characterizing the Run-time Behavior of Stochastic Local Search. Technical Report AIDA–98–1, FG Intellektik, TU Darmstadt, January 1998.

[9] Hoos, H. H; Stützle, T. Evaluating Las Vegas Algorithms — Pitfalls and Remedies. *Proceedings of UAI-98*, pages 238–245. Morgan Kaufmann Publishers, 1998.

[10] Hoos, H. H; Stützle, T. Towards a Characterisation of the Behaviour of Stochastic Local Search Algorithms for SAT. *Artificial Intelligence*, **122:** 213–232, 1999.

[11] Hoos, H. H; Stützle, T. Local Search Algorithms for SAT: An Empirical Evaluation. *To appear: J. Automated Reasoning, Special Issue "SAT 2000"*, 2000.

[12] Kautz, H.; Selman, B. Pushing the Envelope: Planning, Propositional Logic, and Stochastic Search. *Proceedings of AAAI'96*, volume 2, pages 1194–1201. MIT Press, 1996.

[13] Li, C. M.; Anbulagan. Look-ahead versus look-back for satisfiability problems. *Proceedings of CP'97*, LNCS, pages 341–355. Springer Verlag, 1997.

[14] Stützle, T. Local Search Algorithms for Combinatorial Problems – Analysis, Algorithms, and New Applications. infix Verlag, St. Augustin, Germany, 1999..

[15] Walsh, T. Search in a small world. *Proceedings of IJCAI-99*, to appear.