
Journal of Automated Reasoning **24:** 67–100, 2000.

Heavy-Tailed Phenomena in Satisfiability and Constraint Satisfaction Problems

CARLA P. GOMES and BART SELMAN
Dept. of Computer Science, Cornell University, Ithaca, NY 14853, U.S.A.
e-mail: gomes@cs.cornell.edu, selman@cs.cornell.edu

NUNO CRATO
Dept. of Mathematics, ISEG, Technical University of Lisbon, Portugal. e-mail: ncrato@iseg.utl.pt

HENRY KAUTZ
AT&T Shannon Laboratories, Florham Park, NJ 07932, U.S.A. e-mail: kautz@research.att.com

Abstract. We study the runtime distributions of backtrack procedures for propositional satisfiability and constraint satisfaction. Such procedures often exhibit a large variability in performance. Our study reveals some intriguing properties of such distributions: They are often characterized by very long tails or "heavy tails". We will show that these distributions are best characterized by a general class of distributions that can have infinite moments (i.e., an infinite mean, variance, etc.). Such nonstandard distributions have recently been observed in areas as diverse as economics, statistical physics, and geophysics. They are closely related to fractal phenomena, whose study was introduced by Mandelbrot. We also show how random restarts can effectively eliminate heavy-tailed behavior. Furthermore, for harder problem instances, we observe long tails on the left-hand side of the distribution, which is indicative of a non-negligible fraction of relatively short, successful runs. A rapid restart strategy eliminates heavy-tailed behavior and takes advantage of short runs, significantly reducing expected solution time. We demonstrate speedups of up to two orders of magnitude on SAT and CSP encodings of hard problems in planning, scheduling, and circuit synthesis.

Key words: satisfiability, constraint satisfaction, heavy tails, backtracking

1. Introduction

Procedures for solving propositional satisfiability (SAT) problems often exhibit a remarkable variability in the time required to solve any particular problem instance. For example, we see significant differences on runs of different heuristics, runs on different problem instances, and, for stochastic methods, runs with different random seeds. The inherent exponential nature of the search process appears to magnify the unpredictability of search procedures. In fact, it is not uncommon to observe a satisfiability procedure "hang" on a given instance, whereas a different heuristic, or even just another stochastic run, solves the instance quickly.

The fastest complete procedures for SAT are based on the Davis–Putnam–Logemann–Loveland method [12, 13]. These procedures essentially perform a backtrack search through the space of truth assignments. We study the runtime distributions of such backtrack search methods on a variety of problem instances. As we will see, these distributions have a number of intriguing properties. In particular, the distributions are often characterized by very long tails or "heavy tails". We will show that these distributions are best captured by a general class of distributions that can have infinite moments. Aside from our experiments on SAT, we also demonstrate the heavy-tailed phenomenon for backtrack search on general constraint satisfaction problems. In fact, we believe our results apply to backtrack search techniques in general.

Heavy-tailed distributions were first introduced by the Italian-born Swiss economist Vilfredo Pareto in the context of income distribution. They were extensively studied mathematically by Paul Lévy in the period between the world wars. Lévy worked on a class of probability distributions with heavy tails, which he called *stable* distributions. However, at the time, these distributions were largely considered probabilistic curiosities or pathological cases mainly used in counterexamples. This situation changed dramatically with Mandelbrot's work on fractals. In particular, two seminal papers by Mandelbrot [42, 43] were instrumental in establishing the use of stable distributions for modeling real-world phenomena.

Recently, heavy-tailed distributions have been used to model phenomena in areas as diverse as economics, statistical physics, and geophysics. More concretely, they have been applied in stock market analysis, Brownian motion, weather forecasts, earthquake prediction, and recently, for modeling time delays on the World Wide Web (e.g., [1, 44, 56]).

Various researchers studying the computational nature of search methods on combinatorial problems have informally observed the erratic behavior of the mean and the variance of the search cost. In fact, this phenomenon has led researchers studying the nature of computationally hard problems to use the median cost, instead of the mean, to characterize search difficulty, because the median is generally much more stable [9, 20, 27, 35, 46, 64]. More recently, the study of runtime distributions of search methods – instead of just the moments and median – has been shown to provide a better characterization of search methods and much useful information in the design of algorithms [18, 23, 21, 29, 36, 55].

The work reported here provides a characterization of the tail behavior of backtrack-style SAT and CSP procedures, which often dominates their overall performance. We use the probabilistic model of heavy-tailed distributions. Such distributions provide a formal framework explaining the large variance and the erratic behavior of the mean of backtrack search. See Figure 1(a) for a preview of the erratic mean phenomenon. The figure shows the mean cost calculated over an increasing number of runs, on the same problem instance, of a randomized backtrack search procedure on a constraint satisfaction formulation of the quasigroup completion problem (described below). Contrast this behavior with that of the mean of

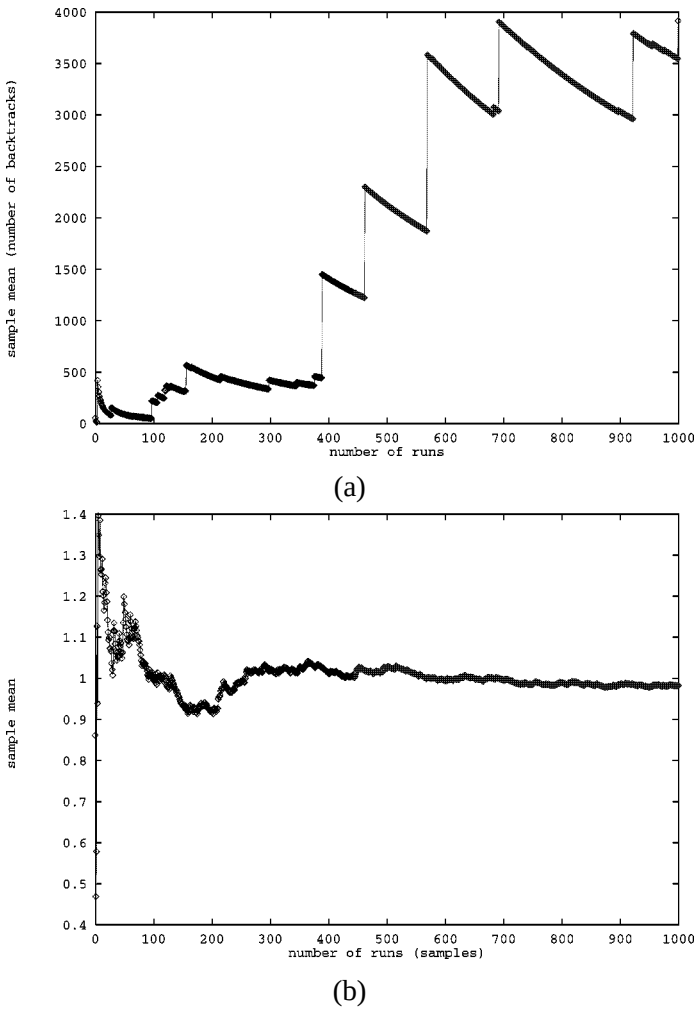


Figure 1. Erratic behavior of the sample mean of backtrack search for completing a quasigroup (order 11, 30% pre-assignment) vs. stabilized behavior of the sample mean for a standard distribution (gamma).

a standard probability distribution (e.g., a gamma distribution; no heavy tails), as given in Figure 1(b). In this case, we see that the sample mean converges rapidly to a constant value with increasing sample size. On the other hand, the heavy-tailed distribution in Figure 1(a) shows a highly erratic behavior of the mean that does not stabilize with increasing sample size.*

Previous authors have discovered the related – and quite surprising – phenomenon of so-called exceptionally hard SAT problems in fixed problem distributions [20, 62]. For these instances, we further observed that when a small amount of

* The median, not shown here, stabilizes rather quickly at the value 1.

randomization was introduced into the heuristic used by the search algorithm, then, on some runs, the instances were solved quickly [58]. Thus, the "hardness" did not necessarily reside only in the instances, but rather in the combination of the instance with the details of the deterministic algorithm. When we plotted the solution times for many runs of the randomized complete algorithm (with different random seeds) on a *single* problem instance, we discovered the same heavy-tailed behavior as we had seen before on a collection of instances.

In our empirical work, we analyze cost distributions of collections of runs on single SAT instances, as well as on ensembles of instances. When running a search procedure on an ensemble of instances, we encounter two sources of variability: One is due to the variability between problem instances and the other one is due to the search algorithm itself. In order to isolate the latter source of variability – to better understand the nature of our search methods – we emphasize the study of distributions obtained from collections of runs (with different random seeds) on single problem instances. By considering instances from a range of problem domains, we also obtain insights into the variability across domains. See [29, 31] for a similar methodology.

As a direct practical consequence of the heavy-tailed behavior of backtrack search methods, we show how *rapid randomized restarts* (RRR) can dramatically reduce the variance in the search behavior. In fact, as we will see, a search strategy with restarts can eliminate heavy-tailed distributions.

For our experiments, we used SAT encodings of known hard problem instances from timetabling, planning, code optimization, and circuit synthesis. We also considered CSP encodings of scheduling problems. For our solvers, we used two state-of-the-art satisfiability engines: Satz by Li and Anbulagan [39] and Relsat by Bayardo and Schrag [5], and an efficient CSP solver built using the Ilog C++ constraint programming library [53]. It is important to note that the underlying deterministic complete search engines are among the fastest (and on many problems, *the* fastest) in their class. Thus, the techniques discussed in this paper extend the range of complete methods to problems that were often previously beyond their reach.

The paper is structured as follows. In the next section, we describe how we randomize Davis–Putnam style search methods and define our problem domains. In the following section, we introduce the notion of heavy-tailed distributions and analyze our empirical data. In the fourth section, we discuss how one can exploit the underlying heavy-tailed distributions to improve satisfiability and constraint satisfaction methods. The final section summarizes our results and gives directions for future research.

2. Search Procedures and Problem Domains

2.1. RANDOMIZATION

We consider a general technique for adding randomization to complete, systematic, backtrack search procedures, such as the Davis–Putnam procedure. These procedures construct a solution (or truth assignment) incrementally. At each step a heuristic is used to select an operation to be applied to a partial solution, such as assigning a value to an unassigned variable. Eventually either a complete solution is found, or the algorithm determines that the current partial solution is inconsistent. In the latter case, the algorithm backtracks to an earlier point in its search tree.

If several choices are heuristically determined to be equally good, then a deterministic algorithm applies some fixed rule to pick one of the operations, for example, by selecting the variables in lexicographic order. The most obvious place to apply randomization, therefore, is in this tie-breaking step: if several choices are ranked equally, choose among them at random. Even this simple modification can dramatically change the behavior of a search algorithm, as we will see below.

However, if the heuristic function is particular powerful, it may rarely assign more than one choice the highest score. To handle this, we can introduce a "heuristic equivalence" parameter to the algorithm. Setting the parameter to a value H greater than zero means all choices that receive scores within H-percent of the highest score are considered equally good. This expands the choice set for random tie-breaking.

With these changes, each run of the search algorithm on a particular instance will differ in the order in which choices are made and potentially in time to solution. We note that introducing randomness in the branching variable selection does not affect the completeness of the backtrack search. Some basic bookkeeping ensures that the procedures do not revisit any previously explored part of the search space, which means that we can still determine inconsistencies, unlike local search methods. The bookkeeping mechanism involves some additional information, where for each variable on the stack, we keep track of which (truth) assignments have been tried so far.

We modified two state-of-the-art SAT solvers, Satz [39] and Relsat [6].* Both procedures are versions of the Davis–Putnam–Logemann–Loveland procedure [12, 13] with sophisticated heuristics for choosing which variable to branch on at each branching point. Relsat incorporates random tie-breaking and lookback strategies, such as conflict directed backjumping (CBJ, [52]) and relevance bounded learning [5]. These procedures are the fastest SAT methods we have found for the instances discussed in this paper. In both procedures the powerful heuristics often only yield a relatively small set of variables to branch on. We therefore added the heuristic equivalence parameter H to enlarge the choice set.

* We thank Chu Min Li and Roberto Bayardo for making their source code available to us. See also SATLIB at www.informatik.tu-darmstadt.de/AI/SATLIB/.

We also randomized the Ilog constraint solver for our experiments on constraint satisfaction formulations. Ilog provides a powerful C++ constraint programming library [53]. We randomized the first-fail heuristic and various variants of the Brelaz selection rule, which has been shown to be effective on graph-coloring-style problem domains [7].

2.2. PROBLEM DOMAINS

In our study, we consider a series of problem domains.* Our first domain is the so-called quasigroup completion problem. This domain was introduced in Gomes and Selman [22] to bridge the gap between purely random instances and highly structured problems, such as those from finite algebra [19, 37, 61].

A quasigroup is an ordered pair $(Q, \cdot)$, where Q is a set and $(\cdot)$ is a binary operation on Q such that the equations $a \cdot x = b$ and $y \cdot a = b$ are uniquely solvable for every pair of elements a, b in Q. The *order* N of the quasigroup is the cardinality of the set Q. The best way to understand the structure of a quasigroup is to consider the N by N multiplication table as defined by its binary operation. The constraints on a quasigroup are such that its multiplication table defines a *Latin square*. This means that in each row of the table, each element of the set Q occurs exactly once; similarly, in each column, each element occurs exactly once [14].

An *incomplete* or *partial Latin square* P is a partially filled N by N table such that no symbol occurs twice in a row or a column. The quasigroup completion problem is the problem of determining whether the remaining entries of the table can be filled in such a way that we obtain a complete Latin square, that is, a full multiplication table of a quasigroup. We view the pre-assigned values of the Latin square as a *perturbation* to the original problem of finding an arbitrary Latin square. Another way to look at these pre-assigned values is as a set of additional problem constraints on the basic structure of the quasigroup. (See www.cs.cornell.edu/gomes/ for a Java applet demonstrating the quasigroup completion problem.)

The quasigroup completion problem is NP-complete [3, 8]. In previous work, we identified a phase transition phenomenon for the completion problem [22]. At the phase transition, problem instances switch from being almost all solvable ("underconstrained") to being almost all unsolvable ("overconstrained"). The computationally hardest instances lie at the phase transition boundary. This phase transition allows us to tune the difficulty of our problem class by varying the percentage of pre-assigned values. The location of the phase transition for quasigroups of order up to around 20 occurs around 42% of pre-assigned colors. For larger quasigroups, the exact location of the phase transition appears dependent on the order of the quasigroup, which is a topic for further study.

Figure 2 shows the median computational cost and phase transition as functions of the percentage of pre-assigned values for quasigroups up to order 15. Each data point is generated using 1,000 problem instances. The pre-assigned values were

* Problem instances and data are available from the first author (gomes@cs.cornell.edu). All experiments were performed on a 194 MHz SGI R10000 processor.

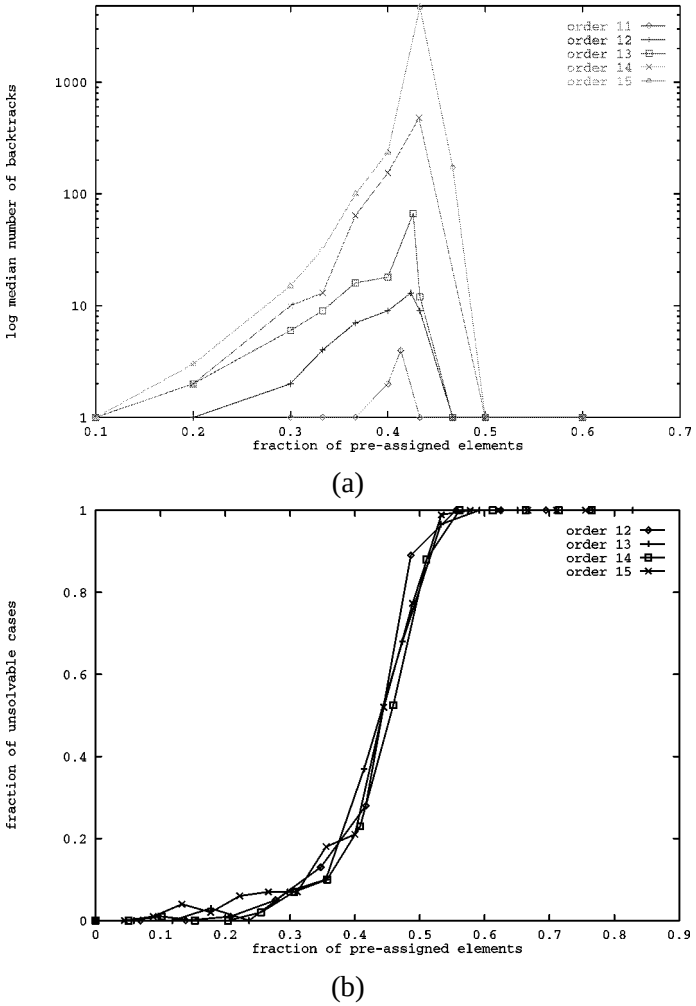


Figure 2. (a) Cost profile, and (b) phase transition for the quasigroup completion problem (up to order 15).

randomly generated in a incremental fashion with forward checking to eliminate obvious conflicts. For interesting related work using a somewhat different model of pre-assignment, see [59].

In our experiments, we used a constraint satisfaction formulation (CSP) formulation of the quasigroup completion problem. We did some preliminary experiments with a satisfiability (SAT) encoding but found that the formulas grow very large rather quickly, making it difficult to study quasigroup problems of interesting size. However, our preliminary SAT experiments agreed with our experiments using the CSP encoding.*

* We thank Mark Stickel for randomizing his implementation of the Davis–Putnam procedure and for providing us with a SAT generator for the quasigroup problem.

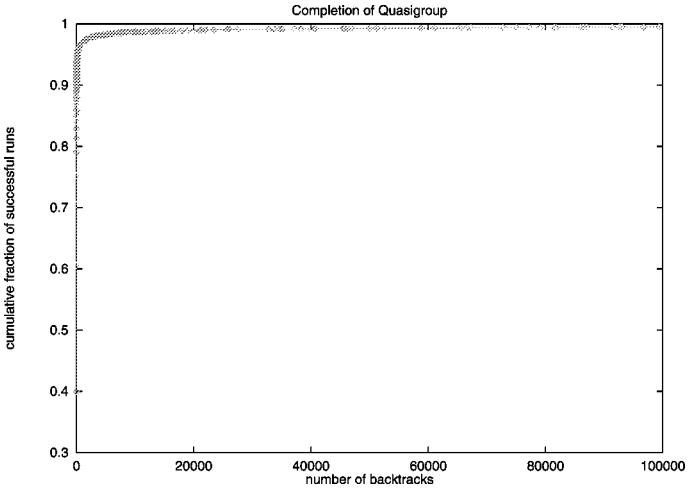
Our second problem domain is from the area of timetabling. In this domain, the problem is to determine whether there exists a feasible schedule that takes into consideration a set of pairing and distribution constraints. We consider two types of timetabling problem domains: school timetabling ([32]; we used a SAT encoding), and round-robin sports team scheduling ([45]; we used a CSP encoding). For some of the background literature in this fast-growing area and to get a sense of the range and mathematical difficulty of the problems encountered, see, for example, [48].

Our third problem domain is planning. Kautz and Selman [34] showed that propositional SAT encodings of STRIPS-style planning problems could be efficiently solved by SAT engines. While both a complete backtrack-style engine and an incomplete local search engine worked well on moderate-sized problems, the largest problems from the domain of logistics scheduling could be solved only by local search. However, as it turns out, the deterministic version of Satz can solve all of the logistics instances from that paper in less than 2 minutes. Therefore we constructed a still-larger planning problem, labeled "logistics.d". This domain involves moving packages on trucks and airplanes between different locations in different cities. While the largest logistics problem from the Kautz and Selman [34] paper involved 1,141 variables and 10^{10} states, "logistics.d" involves 2,160 variables and 10^{16} states. In the logistics domain, a state is a particular configuration of packages and vehicles. Satz takes over 100 minutes on logistics.d.

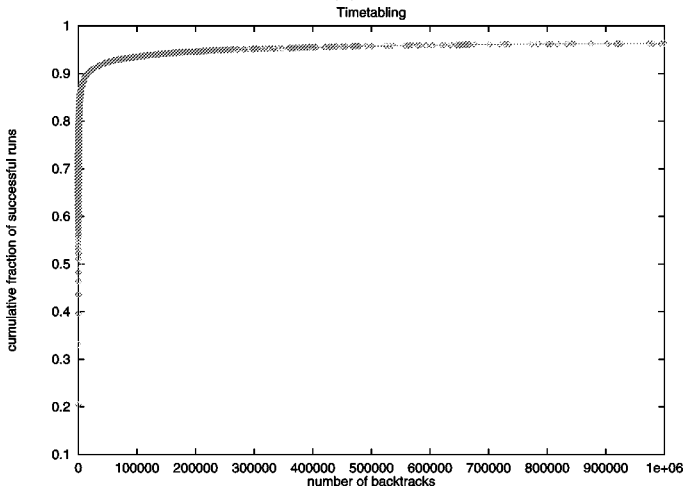
Our fourth and final domain involves several SAT encodings of problems from the Dimacs Challenge benchmark [32]. We consider a code optimization problem, involving register allocation ("mulsol" instance), and circuit synthesis problems ("adder" instances). For the circuit synthesis problems, Kamath et al. [33] developed a technique for expressing the problem of synthesizing a programmable logic array (PLA) as a propositional satisfiable problem. The statement of the problem includes a table specifying the function to be computed, and an upper bound on the number of gates that may appear in the circuit. In general, these problems become more difficult to solve as the number of gates is reduced, until the limit is reached where the instance becomes unsatisfiable. These problems are quite hard to solve with complete SAT procedures and have been used as part of the test beds for numerous SAT competitions and research studies. The problems considered in this paper, "3bit-adder-32" and "3bit-adder-31", are (as one would guess) based on synthesizing a 3-bit adder using 32 and 31 gates, respectively. Although Selman et al. [57] solve the instances using local search, they have not previously been solved using a backtrack procedure.

3. Cost Distributions of Backtrack Search

As mentioned in the introduction, our first experiments demonstrate some of the puzzling features of backtrack search procedures, such as an extreme variability and a seemingly "wandering sample mean" of the search cost. See Figure 1(a). The data is based on runs of a randomized backtrack search procedure (Ilog) on an



(a)



(b)

Figure 3. Long tails for (a) quasigroup completion (CSP formulation), and (b) timetabling (SAT formulation).

instance of the quasigroup problem of order 11 with 30% of pre-assignment. The figure shows the highly erratic behavior of the mean, which does not stabilize with increasing sample size. On the other hand, as we noted earlier, the median is 1 and stabilizes rather quickly.

Figure 3 provides some insights into the cause of the "wandering mean phenomenon." The figure shows surprisingly long "tails" in the distributions for two experimental domains. Each curve gives the cumulative fraction of successful runs as a function of the number of backtracks, and was produced by running the randomized backtrack search procedure 10,000 times on the same instance. Figure 3(a) shows

the performance profile for the quasigroup completion problem; the data is from the same instance as shown in Figure 1. Figure 3(b) shows the cost profile of a randomized search procedure for a school timetabling problem ("school1_nsh" [32]).

The long-tail phenomenon is apparent in both curves. In the case of the quasigroup completion problem, even though 50% of the runs solve the instance in 1 backtrack or less, after 100,000 backtracks 0.5% of the runs were still not completed. For the school timetabling problem, in 80% of the runs, a solution is found in 1,000 backtracks or less. However, 5% of the runs do not result in a solution even after 1,000,000 backtracks. (Note that, given that our procedure is complete, each of these runs would eventually find a solution.)

In order to model the long-tail behavior of our distributions, we will consider nonstandard probability distributions, referred to as "heavy-tailed" distributions. These distributions have recently received much attention because of their suitability to model stochastic phenomena subject to extreme fluctuations. In the next section, we provide a brief introduction to heavy-tailed distributions and some of their mathematical properties. In the following section, we show how the long tails of the runtime distributions of backtrack search methods can be captured using the heavy-tailed model.

3.1. HEAVY-TAILED DISTRIBUTIONS

Standard probability distributions, such as the normal distribution, have exponentially decreasing tails, which means that events that are several standard deviations from the mean of the distribution ("outliers") are very rare.* In this section, we consider distributions that have rather different properties, often leading to non-intuitive behavior. More specifically, we consider distributions that asymptotically have "heavy tails" – also called tails of the Pareto–Lévy form, namely,

$$\mathrm{P}\{X > x\} \sim Cx^{-\alpha}, \quad x > 0, \tag{1}$$

where $0 < \alpha < 2$ and $C > 0$ are constants. These are distributions whose tails have a *power-law* decay.

An important source of heavy-tailed distributions is the class of so-called *stable* distributions. Stable distributions have been proposed in the study of several types of physical and economic systems. This class of distributions was extensively studied by Paul Lévy to model phenomena characterized by sums of independent identically distributed random variables. Informally, a random variable X is stable if the distribution of the sum of independent copies of it has the same shape as the distribution of X. The name of these distributions emphasizes the fact that the shape of a stable distribution is *stable* under addition.

* The tail of the standard normal distribution (mean 0 and standard deviation 1) decays as $\mathrm{P}\{X > x\} \sim \frac{1}{x\sqrt{2\pi}}\mathrm{e}^{-x^2/2}$ [16]. We write $h(x) \sim g(x)$ to mean $\lim_{x\to\infty} h(x)/g(x) = 1$.

Formally, a random variable X is *stable* if we have the following relation between X and two copies of X, X_1, and X_2, and any positive constants a and b:

$$aX_1 + bX_2 \stackrel{\mathcal{D}}{=} cX + d \tag{2}$$

for some positive c and $d \in R$. The symbol $\stackrel{\mathcal{D}}{=}$ means equality in distribution; that is, both expressions have the same probability law, possibly with different scale and location parameters [16]. In effect, (2) states that the distribution of the sum of two copies of X behaves essentially the same as X itself (up to a location and scaling factor).

Most standard probability distributions are not stable. For example, although the sum of two independently distributed gamma distribution is still a gamma, the shape of the resulting gamma is generally not preserved. The normal or Gaussian distribution, however, is stable. There are two more cases of stable distributions for which one can write down explicit (closed form) expressions for their probability density functions. These are the Cauchy distribution and the Lévy distribution.⋆ These distributions differ, however, in a fundamental way from the normal distribution in that the Cauchy and the Lévy distributions are heavy-tailed distributions: That is, the tails decay as in Equation (1). Stable distributions are heavy tailed, with the notable exception of the normal distribution [51]. The normal distribution is a stable distribution but not heavy tailed. The other standard distributions, such as the exponential, Weibull, lognormal, and gamma, are neither stable nor heavy tailed. Such standard distributions all have exponentially decaying tails and finite moments.

In Table I, we compare the tail probabilities for the three distributions. It is clear that the tail probability for the normal quickly becomes negligible, whereas the other two distributions have a significant probability mass in the tail.

The α in (1) is referred to as the *index of stability* of the distribution. The lower the index, the heavier the tail. For example, the Cauchy has $\alpha = 1.0$, and the Lévy distribution has $\alpha = 0.5$.

Heavy-tailed distributions have a number of surprising properties. For example, let's consider the moments (mean, variance, etc.) of these distributions. For standard distributions, these moments are well defined. In the case of heavy-tailedness, due to the large probability mass in the tails, some of the integrals that define the moments do not converge. In particular, for $\alpha < 2$, moments of X of order less than α are finite while all higher-order moments are infinite, that is, $\alpha = \sup\{b > 0 : \mathrm{E}|X|^b < \infty\}$. So, when $1 < \alpha < 2$, the distribution has a finite mean but no finite variance. With $\alpha \leq 1$, the distribution has neither a finite mean nor a finite variance.

⋆ For the normal distribution, $X \sim N(\nu, \sigma^2)$, the probability density is given by $f(x) = \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(x-\nu)^2}{2\sigma^2}}$. For the Cauchy, $X \sim \text{Cauchy}(\gamma, \delta)$, we have $f(x) = \frac{1}{\pi} \frac{\gamma}{\gamma^2 + (x-\delta)^2}$. And for $X \sim \text{Lévy}(\gamma, \delta)$, $f(x) = \sqrt{\frac{\gamma}{2\pi}} \frac{1}{(x-\delta)^{3/2}} \mathrm{e}^{-\frac{\gamma}{2(x-\delta)}}$ [51].

Table I. Comparison of tail probabilities, $P\{X > c\}$, for standard normal, Cauchy, and Lévy distributions. (Adapted from [51].)

c	Normal	Cauchy	Lévy
0	0.5000	0.5000	1.0000
1	0.1587	0.2500	0.6827
2	0.0228	0.1476	0.5205
3	0.001347	0.1024	0.4363
4	0.00003167	0.0780	0.3829
5	0.0000002866	0.0628	0.3453

Many aspects of random walks involve heavy-tailed distributions [16, 17]. Consider a one-dimensional random walk, where at each time step one takes a unit-step to the left or right with equal probability. One can show that after starting at the origin, with probability one, the walk will eventually return to the origin. However, the *expected* time before return is infinite, and, on average, the walk will reach all values on the x-axis before its return. Another intriguing phenomenon involves the expected number of returns to the origin ("zero-crossings") in a given number of steps. Intuition would dictate that if in k steps one has on average l crossings, then in a walk that is m times as long, one would expect on average $m \times l$ crossings. However, it can be shown that in $m \times l$ steps, one will observe, on average, only $\sqrt{m} \times l$ crossings. This means that there can be surprisingly long periods in a walk between two crossings. In fact, in a series of r random walks, each terminating at the first crossing, on average, some of the walks will be of the same order as the length of all other walks combined, *no matter what the value of r is.* Such events would normally be dismissed as outliers, but with heavy-tailed distributions, they are far from rare and are an inherent aspect of the distribution. These distributions are therefore good models for dealing with phenomena that exhibit extreme fluctuations.

To provide a visual illustration of the heavy-tail effect in random walks, consider Figure 4. The figure shows simulation data for 10,000 runs of a symmetric random walk. For each walk we recorded the number of steps before the walk returned to the origin. In the figure, we plot the complement-to-one of the cumulative distribution, that is, $1 - F(x) = 1 - P\{X \leq x\} = P\{X > x\}$, with $F(x)$ being the cumulative distribution of $f(x)$. The probability function, $f(x)$, gives the probability of returning to the origin in exactly x steps. So, $1 - F(x)$ gives us the probability of returning in more than x steps. It can be seen that $f(1) = 0$ and $f(2) = 0.5$, so we have $F(x) = 0.5$, which means that with 50% chance the walk returns to the origin in at most two steps. In the figure, we give the log-log plot for $1 - F(x)$. (In the figure, the walk data is given by the diagonal straight line.) As we can see, we obtain a nearly straight line for the tail of the distribution. This suggests that the

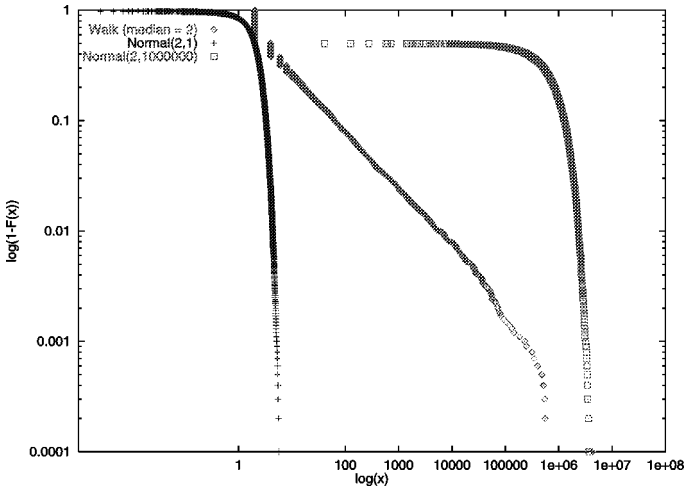


Figure 4. The heavy-tailed nature of a random walk with the exponential decay of the normal distribution for comparison.

$1 - F(x)$ has power law decay, that is, we have $1 - F(x) = \mathrm{P}\{X > x\} \sim Cx^{-\alpha}$, and thus the distribution is heavy-tailed according to our definition (1) above. The slope of the line gives us an estimate of the index of stability, α, which in this case is equal to 0.5 (also known by rigorous analysis). The relatively high frequency of large outliers is clear from the figure. For example, although 50% of the walks return in just 2 steps or less, 1% of the walks take more than 5,000 steps to return to the origin, and about 0.1% take over 200,000 steps. In fact, several of the walks in our sample take almost 1,000,000 steps.

To demonstrate how different such a heavy-tailed distribution is from a standard distribution, we included simulation data of the complement-to-one of the cumulative distribution for a normal distribution. We used a mean value of 2 and give the curves for two different standard deviations ($\sigma = 1$, left-most curve, and $\sigma = 10^6$, right-most curve). The key property to observe is the sharp, faster-than-linear decay of the normal distribution in the log-log plot, which is consistent with the exponential decay in the tail of the distribution. We included a normal distribution with $\sigma = 10^6$ to show that the drop-off of the tail remains sharp even when the normal distribution has a large standard deviation. (The normal distribution is symmetrical; the figure gives only the right-hand side.)

There is a substantial literature on heavy-tailed distributions. Mandelbrot in [44] provides a good introduction to these distributions, with a discussion of their inherently self-similar or fractal nature. For a complete treatment of stable distributions, see either [66] or the more modern approach of [56].

3.2. EMPIRICAL RESULTS

In order to check for the existence of heavy tails in our runtime distributions for backtrack search, we proceed in two steps. First, we graphically analyze the tail

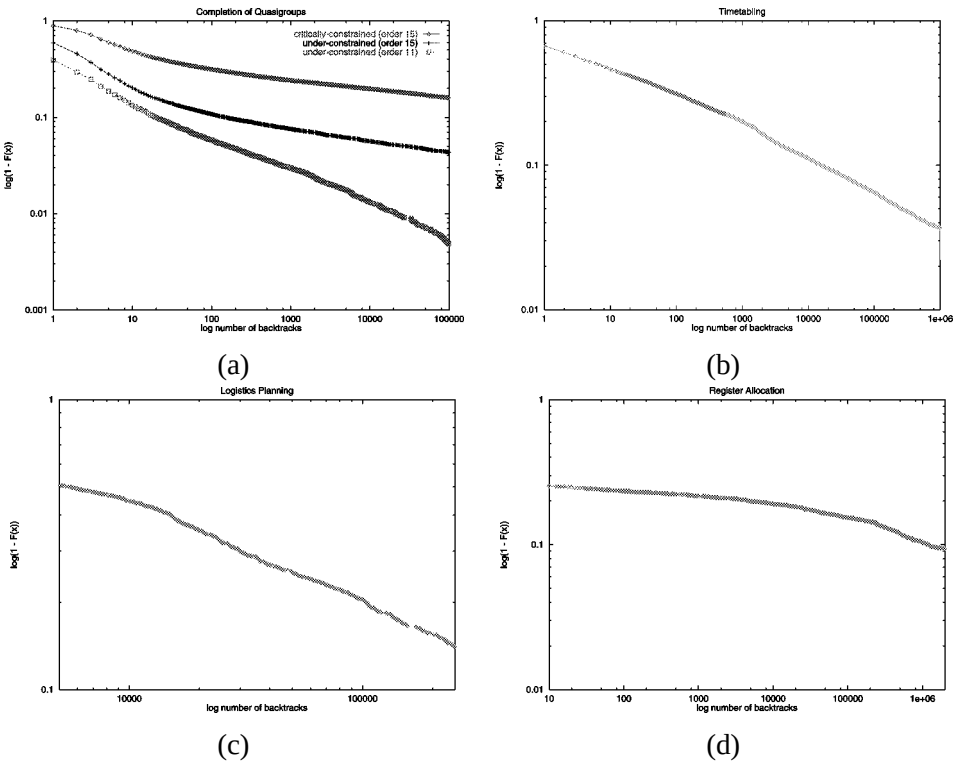


Figure 5. Log-log plot of heavy-tailed behavior for (a) quasigroup completion (CSP formulation); (b) timetabling (SAT formulation); (c) logistics planning (SAT formulation), and (d) register allocation (SAT formulation).

behavior of the sample distributions. Second, we formally estimate the index of stability.

From (1), we have $1 - F(x) = \mathrm{P}\{X > x\} \sim Cx^{-\alpha}$. As in the case of the random-walk example, given the power law decay of $1 - F(x)$, its log-log plot should show an approximate linear decrease in the tail. Moreover, the slope of the curve provides an estimate of the index α. In contrast, for a distribution with an exponentially decreasing tail, the log-log plot should show a faster-than-linear decrease.

Figure 5 displays the log-log plots of the complement-to-one of the cumulative distributions for our experimental domains. In Figure 5(a), we give the data for a solvable critically constrained instance of the quasigroup completion problem (order 15, 40% pre-assignment), and two solvable underconstrained instances of the quasigroup completion problem (one of order 15, 30% pre-assignment and the other of order 11, 30% pre-assignment, the same instance as used in Figures 1(a) and 3(a)). Figure 5(b) gives the data for the school timetabling problem; 5(c) for the logistics planning problem; and 5(d) for the register allocation problem. The linear nature of the tails in these log-log plots directly reveals the heavy-tailed behavior.

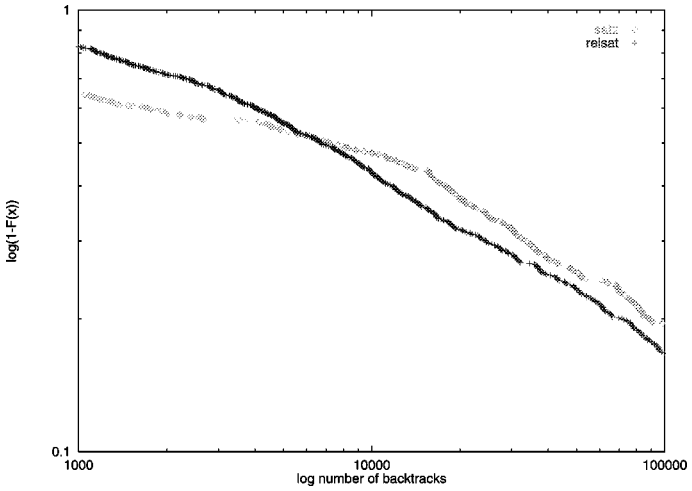
Note that our data is based on solvable problem instances. We considered additional distributions of over two dozen randomly picked quasigroup completion problems, from both the underconstrained and the critically constrained area, and other instances from the domains described above, as well as some ensembles of instances at various degrees of constrainedness. For sufficiently large instances (and therefore requiring substantial search), we found heavy-tailed distributions for most of our solvable instances. Some easy solvable instances did not exhibit heavy tails. We also have not encountered inconsistent instances with heavy-tailed behavior, which is consistent with the work on unsolvable instances by [18].

Figure 6(a) shows a comparison between two different SAT procedures on the logistics planning problem. We compare Satz [39] with Relsat [6]. As mentioned earlier, both procedures use sophisticated propagation rules and heuristics to guide their search. However, the Relsat backtracking strategy also includes look-back, based on conflict directed backjumping and relevance bounded learning [52, 5]. We present this comparison to consider the possible influence of such look-back strategies on the heavy-tailed behavior. From the figure, we see that Relsat still results in heavy-tailed behavior but the slope of the distribution in the log-log plot is steeper than for Satz. This means the index of stability for Relsat is higher, meaning a tail that is somewhat less heavy. This observation is consistent with related work where look-back strategies were shown to be effective for solving so-called exceptionally hard SAT instances [6]. In effect, the look-back approach reduces the variability in the search, even though it may not necessarily eliminate it.

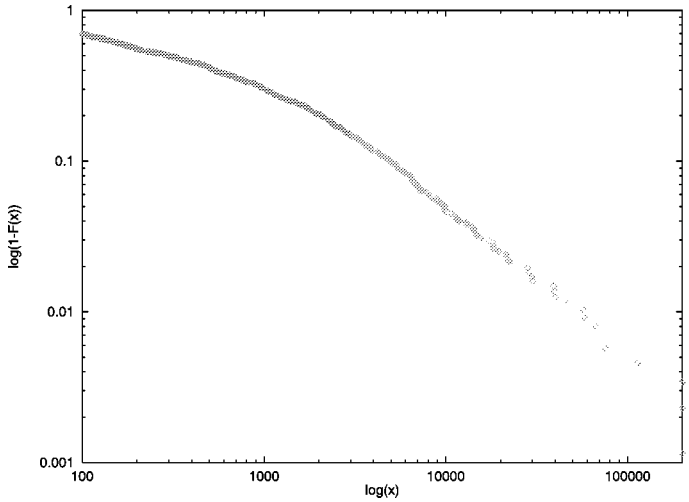
We now consider the effect on heavy-tailed behavior of more compute-intensive propagation methods. For the quasigroup completion problem, when using so-called "all-different-constraints", one can solve substantially larger problems, as discovered by Shaw et al. [59]. In this approach, one maintains general arc consistency on the N-ary all different constraints using the algorithm of Regin [54]. This method uses a matching algorithm to ensure that at each point during the search, in each row, the set of remaining colors for the yet to be assigned cells is such that there is a different color available for each cell (similarly for the columns). Using an efficient Ilog-based implementation, we can solve instances of order 30 in under five minutes, which allows us to gather runtime information within a reasonable time frame.[⋆]

By using the all-different-constraint method, we observed that the heavy-tailed phenomenon disappears for quasigroups of low orders. The reason for this is that the instances are solved mostly by propagation, almost without any search. However, as we increase the order of the quasigroups, the heavy-tail phenomenon reappears. Figure 6(b) shows the heavy tail for an instance of order 30 with 55% preassigned when using the all-different-constraints (55% is close to the phase transition for order 30).

[⋆] We thank Jean Charles Regin for his highly efficient Ilog implementation of this propagation strategy. A similar propagation, more efficient but somewhat less powerful, can be obtained by using so-called conjugate constraints [63].



(a)



(b)

Figure 6. (a) Comparing Satz with Relsat on logistics planning (SAT formulations); (b) heavy tail for quasigroup completion with extensive propagation (CSP formulation).

To complement our visual check of heavy-tailed behavior, we have calculated maximum likelihood estimates of the indices of stability (the values of α). There is a standard estimator for the index, called the Hill estimator [26, 24]. We adapted this estimator, since some of our data is not observable because we ran our experiments with a certain high cutoff (usually 10^6) for the maximum number of backtracks, to avoid literally getting stuck in the most extreme outliers in the tail of the distribution during our data collection.

Table II. Estimates of the index of stability, α, with sample size k. The values within parentheses are the estimated asymptotic standard deviations. (Quasigroup instances encoded as CSP; other instances encoded as SAT.)

Cases	k	α
Quasigroup (order 11, 30%)	9731	0.466 (0.009)
Quasigroup (order 15, 30%)	10000	0.319 (0.006)
Quasigroup (order 15, 40%)	10000	0.153 (0.003)
Quasigroup (order 30, 55%)	10000	0.392 (0.007)
School Timetabling	10000	0.219 (0.004)
Logistics Planning (Satz)	1442	0.360 (0.017)
Logistics Planning (Relsat)	1000	0.670 (0.009)
Register Allocation	10000	0.102 (0.002)

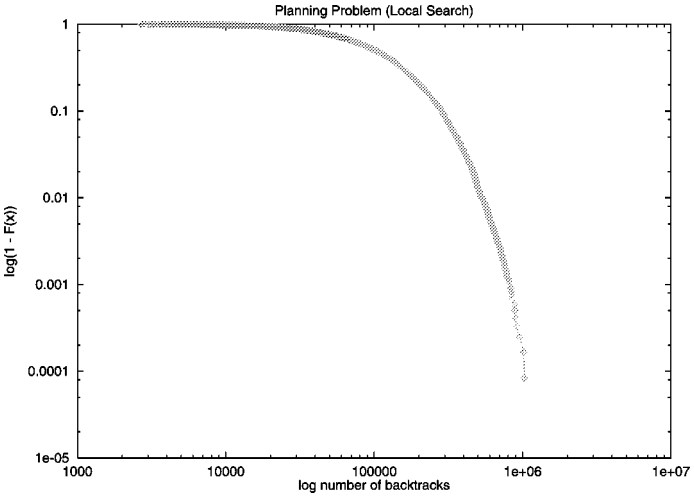
As shown in Appendix A, following an argument similar to the one in [26], we can derive the following maximum likelihood estimator for the index of stability α, which takes into account the data truncation:

$$\hat{\alpha}_{r,u} = \left(\frac{1}{r} \sum_{j=1}^{r-1} \ln X_{n,n-r+j} + \frac{u+1}{r} \ln X_{n,n} - \frac{u+r}{r} \ln X_{n,n-r} \right)^{-1} \tag{3}$$

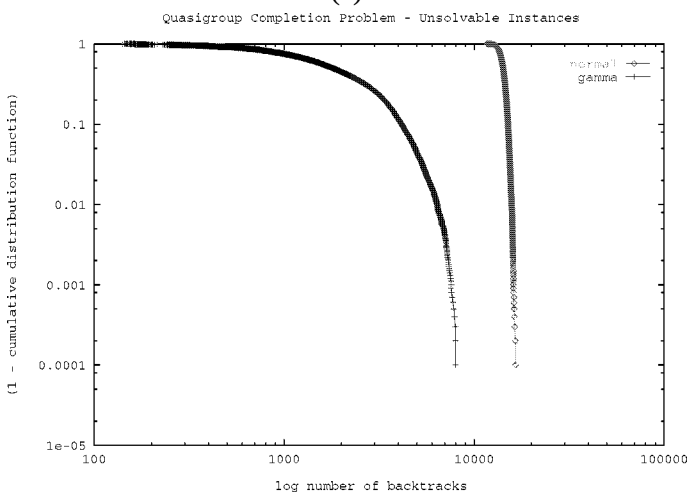
$X_{n,1} \leq X_{n,2} \leq \cdots \leq X_{n,n}$ give the order statistics, that is, the ordered values of the sample $X_1, X_2 \ldots, X_n$ of the observed number of backtracks for different runs of our procedure. The highly extreme variable values $X_{n,n+1} \leq X_{n,n+2} \leq \cdots \leq X_{n,n+u}$ are not observable because of the limit that we imposed on the maximum number of backtracks allowed in the experiments. We used a limit of 10^6. The parameter r ($< n$) is a lower-bound cutoff on search cost, which enables us to focus only on the tail end of the distribution, that is, observations $X_{n,n-r} \leq X_{n,n-r+1} \leq \cdots \leq X_{n,n}$. Without a practical limit on extreme values ($u = 0$), the estimator reduces to the Hill estimator. In our analysis, we considered tails that correspond to roughly 30% of each sample (i.e., $r = 0.3k$, with k the size of the sample).

Table II displays the maximum likelihood estimates of the indices of stability (the values of α) for the instances shown in the various figures. Note that for all the instances shown in the table, the estimates of α are consistent with the hypothesis of infinite mean and infinite variance, since $\alpha < 1$.*

* Of course, the computational cost of complete backtrack algorithms has a finite upper bound. Technically speaking, we are dealing with "truncated heavy-tails". That is, when allowing for a very large number of backtracks such that one can explore most of the complete search tree, one will observe truncation effects in the heavy-tailed distribution. (Personal communication, Holger Hoos, Dec., 1999.) Nevertheless, the heavy-tailed model provides a good approximation for the tail-behavior of the search cost when dealing with realistic-size NP-complete problems. The heavy-tailed model emphasizes the power law decay over several orders of magnitude of the tail of the distribution. Also, since the upper-bound of the search space is exponential in the size of the problem, it is generally not reached in practice.



(a)



(b)

Figure 7. No heavy tails: (a) local search (SAT formulation) and (b) unsolvable instances (CSP formulation).

For contrast with our heavy-tailed results, in Figure 7(a), we show the log-log plot for the same instance from the logistics planning domain used in Figure 5(c), but rather than using a randomized complete search method, we used Walksat [57], a randomized local search procedure. It is clear from that plot that the distribution does not exhibit heavy-tailed behavior, given the faster-than-linear decrease of the tail. This observation is consistent with the recent (independent) results by Hoos [29], showing that the exponential distribution generally provides a good model for the runtime profile of local search.

Another example of nonheavy-tailed behavior is given in Figure 7(b), which shows the log-log plots of two unsolvable instances from the quasigroup completion domain. We see the sharp rounded drop-off of both curves – indicating the absence of heavy tails. One is a rare unsolvable instance in the underconstrained area (best fit: a gamma distribution); the other is an unsolvable instance in the critically constrained region (best fit: normal distribution). As noted above, we have not found unsolvable instances with heavy-tailed behavior, again consistent with Frost et al. [18].

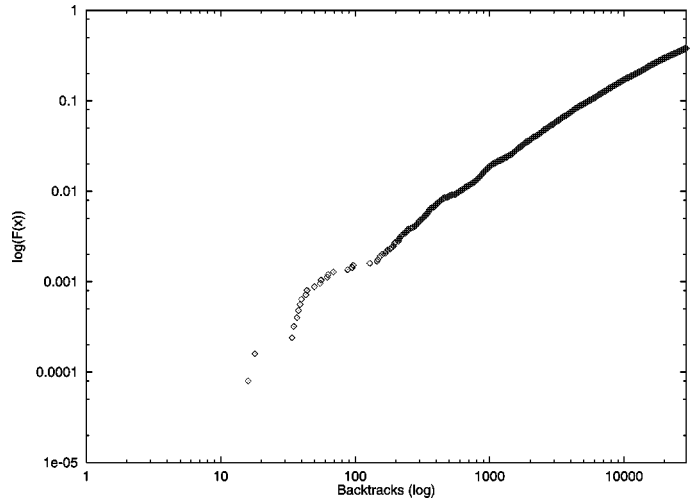
3.3. HEAVY-TAILS ON THE LEFT-HAND SIDE OF THE DISTRIBUTION

The heavy-tailed behavior we have seen so far shows that in backtrack search, extremely long runs can occur much more often than one would expect if the process were covered by a standard probability distribution. Similarly, in further experiments, we have found an indication that, when dealing with relatively hard problem instances, short runs may also occur much more frequently than expected. Figure 8 gives the log-log plot of the cumulative distribution ($F(x) = \mathrm{P}\{X \leq x\}$) for a 14-team round-robin scheduling problem (panel a) and for a logistics planning problem (panel b). The data from Figure 8(a) shows that even though 70% of the runs take more than 25,000 backtracks, roughly 1% of the runs take fewer than 650 backtracks. In the case of the 16-team scheduling instance, we observed that roughly 1% of the runs takes fewer than 10,000 backtracks, compared with a median value of the distribution, which is over 2,000,000. For the logistics problem in Figure 8(b), we found that roughly 2% of the runs find a solution in at most 200 backtracks, while 70% of the runs take more than 100,000 backtracks. Figure 8 reveals a substantial fraction of very short runs. However, the left-hand side tails in Figure 8 are not heavy enough to qualify as "heavy". Nevertheless, our conjecture is that very hard instances may indeed exhibit heavy-tailed behavior on the left-hand side. We are currently investigating such a conjecture. The existence of a considerable mass of probability on the left-hand side of the distribution, as well as the heavy-tailed phenomena on the right-hand side, have significant consequences for algorithmic strategies.* In particular, we will show in the next section how, by rapid restarting a randomized backtrack procedure, one can take advantage of the left-hand side probability mass and eliminate the heavy-tailed behavior on the right-hand side.

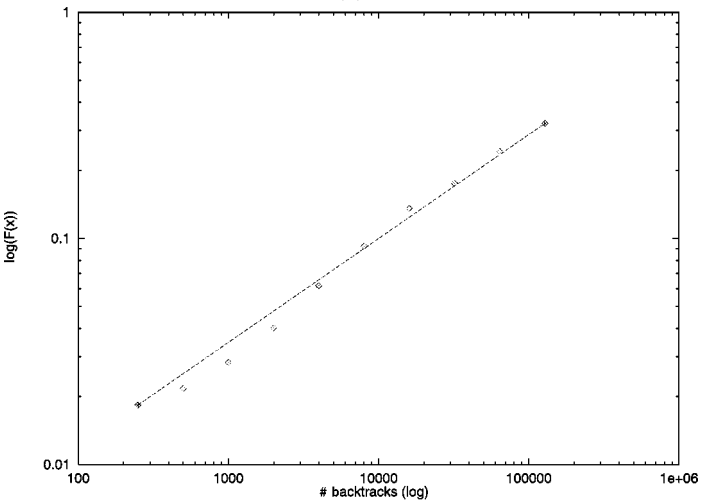
3.4. HEAVY-TAILEDNESS AND STRUCTURE OF SEARCH SPACE

Why do heavy-tailed distributions occur in backtrack search? An intuitive explanation involves the notion of *critically constrained* variables. In most combinatorial

* The long tails on the left also provide an explanation for the so-called superlinear speedup phenomenon, observed in practice when solving hard combinatorial problems using several parallel runs [15, 41, 28, 60]. Basically, a long tail on the left implies that some of the parallel runs have a non-negligible probability of finishing early.



(a)



(b)

Figure 8. Log-log plot of left-hand side tail of the distribution for (a) 14-team round-robin scheduling (CSP formulation) and (b) logistics.d planning instance (SAT formulation).

problems, it is much harder to find good values for some of the variables than for others. However, once these *critical* variables are set, the values for the other variables are largely determined. For example, in scheduling problems it is often the case that once a subset of critical tasks is scheduled, the other tasks immediately fall into place. The most efficient backtrack search strategy is to first concentrate on finding a consistent set of values for these critical variables. The variable choice heuristic employed in backtrack search tries to identify such critical variables, but is in practice inaccurate. The very short runs occur when the heuristic works well

and critical variables are chosen at the beginning of the search. The very long runs occur when the algorithm makes some unfortunate choices early on and explores an exponential number of ways of setting noncritical variables, none of which allows a consistent assignment to the critical variables; that is, the search explores an exponentially large subtree containing no solutions. We are currently formalizing this intuitive explanation of critical variables by studying the close connection between phase transition phenomena and computationally hard problems, using models from statistical physics [25, 35, 47]. We show that the number of critical variables is linked to the nature of the phase transition underlying the computational task. Furthermore, heavy-tailed behavior suggests that the underlying search space for combinatorial problems is inherently self-similar in nature. We are exploring more direct measures to characterize such self-similarity geometrically. Intuitively speaking, self-similarity arises from repetitive patterns in the search tree, leading to a possible fractal dimension of the overall search tree.

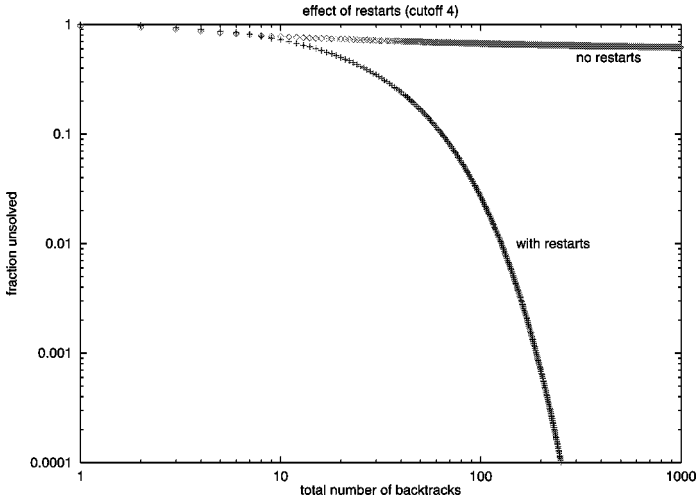
Another step toward a further understanding of heavy-tailed behavior can be found in recent work by Walsh [65]. In searching graphs with so-called "small-world" properties, Walsh [65] identifies heavy-tailed behavior. He puts forward the conjecture that a small-world topology will induce heavy-tailed behavior. A small-world topology is characterized by a small number of global constraints combined with a large number of local connections. In such problems, the long-range interactions may define the critical variables of the problem. Clearly, more work is needed to clarify these issues.

4. Consequences for Algorithm Design

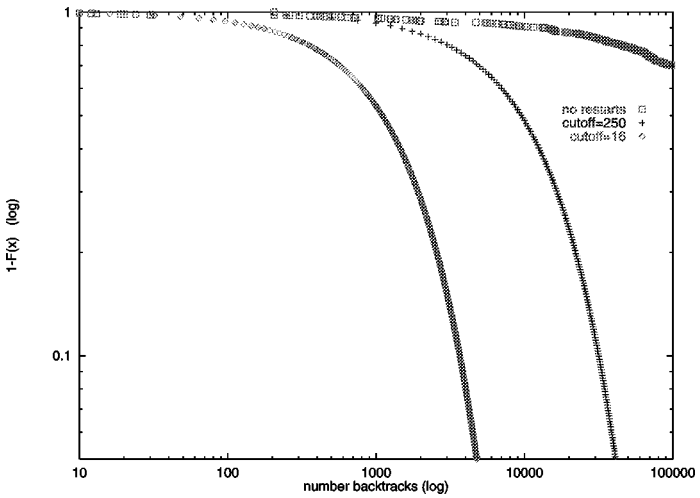
We have shown the special heavy-tailed nature of the cost distributions of randomized SAT and CSP procedures, and we have seen how the distributions can be modeled with Pareto–Lévy type (heavy) tails. Our estimates for α are consistent with the hypothesis of infinite variance ($\alpha < 2$), and infinite mean ($\alpha < 1$), which is consistent with the empirically observed erratic behavior of the mean and extreme variance of the cost of backtrack search methods. We have also found long tails on the left-hand side of the cost distribution for certain classes of problem instances. Given the heavy-tailed phenomenon, a randomized backtrack procedure is, in a sense, most effective early on in the search, which suggests that a sequence of short runs instead of a single long run may be a more effective use of computational resources. In this section, we show how a *rapid randomized restart* strategy can take advantage of the heavy-tailed behavior.

4.1. COST DISTRIBUTIONS WITH RESTARTS

In Figure 9, we show the result of applying a strategy of fixed-length short runs ("restarts") of a randomized backtrack procedure. Figure 9(a) shows the results on a quasigroup completion instance of order 20 with 5% pre-assignment. Without



(a)



(b)

Figure 9. A rapid restart strategy to speed up backtrack search. Failure rate $(1 - F(x))$ as a function of the total number of backtracks for a quasigroup instance (CSP formulation; panel (a)) and a logistics instance (SAT formulation; panel (b)). In panel (b), the leftmost curve is for a cutoff value of 16; the middle curve for a cutoff of 250; and, the rightmost curve is without restarts.

restarts and given a total of 300 backtracks, we have a failure rate of around 70%. With restarts (every 4 backtracks), this failure rate drops to around 0.01%. The figure also shows a clear downward curve for the log-log plot of the complement-to-one of the cumulative distribution of the restart strategy, which is an indication that the heavy-tailed nature of the original cost distribution has disappeared. In fact, we can show that the restart strategy follows a geometric distribution and,

therefore, does not have heavy tails. For a derivation of this result and a formal characterization of restarts, see Appendix B.

Figure 9(b) shows the effect of restarts on the logistics.d planning problem. The figure gives the distributions resulting from running randomized Satz with a cutoff of 16 (near optimal) and a cutoff of 250. The sharp dropoff in the log-log plot shows again the resulting geometric distribution. We see, for example, that with restarts using a cutoff value of 16, after a total of around 5,000 backtracks, we obtain a failure rate of around 5%. On the other hand, *without restarts*, even after 100,000 backtracks, the failure rate is still only around 70%.

4.2. RAPID RANDOMIZED RESTARTS

A restart strategy clearly eliminates the heavy tails on the right of the cost distributions. Perhaps more surprisingly, such restarts can also take advantage of the long tails to the left of the median (Section 3.3). In general, a restart strategy exploits *any significant probability mass early on in the distribution.*

Different cutoff values will result in a different overall mean solution time. Table III shows the mean solution time (based on 100 runs)* for a range of cutoff values on a timetabling problem (panel a, round-robin scheduling; 16 teams), and the logistics.d planning problem (panel b). Both parts of the table show the same overall pattern, revealing a clear optimal range of cutoff values.** For the timetabling problem, the mean cost is minimized with a cutoff around 50,000, and for the planning problem the optimal cutoff value is near 16. (For both instances, these results exploit the long tail on the left.) With a higher than optimal cutoff value, the heavy-tailed behavior on the right of the median starts dominating the overall mean, whereas for cutoffs below the optimal range the success rate is too low, requiring too many restarts to give a good overall mean cost value.‡

Figure 10 gives the data from Table III(b) in a graphical form. The logarithmic vertical scale indicates that one can shift the performance of the procedure over several orders of magnitude by tuning the cutoff parameter.

---

* Note that as a result of the geometric nature of the underlying restart strategy, the standard deviation in the runtime is of the order of the mean.

** It should be noted that random restarts involving a cutoff value are regularly used for local search methods. The novelty here is in the use of restarts in randomized backtrack search to eliminate the heavy-tailed phenomena and to exploit short runs. Restarts are used in local search to avoid local minima, but the underlying runtime distributions do not appear to be heavy tailed (Figure 7 [29]), resulting in less dramatic speedups.

‡ Note that the long tails in Figure 8 do not extend all the way to zero. This suggests that a pure "probing" strategy, that is, repeatedly going down a random branch of the search tree without any backtracking, is not effective on these instances. This also is apparent from Table III. So, at least a small number of backtracks is required. This is in contrast with the scheduling problems considered by Crawford and Baker [11], who obtain good results with a pure probing strategy on SAT encodings in their scheduling domain.

Table III. Solving (a) a 16-team round-robin scheduling problem (CSP formulation) and (b) the logistics.d instance (SAT formulation) for a range of cutoff values.

Cutoff	Success Rate	Mean Cost ($\times 10^6$)
200	0.0001	2.2
5,000	0.003	1.5
10,000	0.009	1.1
50,000	0.07	0.7
100,000	0.06	1.6
250,000	0.21	1.2
1,000,000	0.39	2.5

(a)

Cutoff	Success Rate	Mean Cost
2	0.0	>300,000
4	0.00003	147,816
8	0.0016	5,509
16	0.009	1,861
32	0.014	2,405
250	0.018	13,456
128000	0.32	307,550

(b)

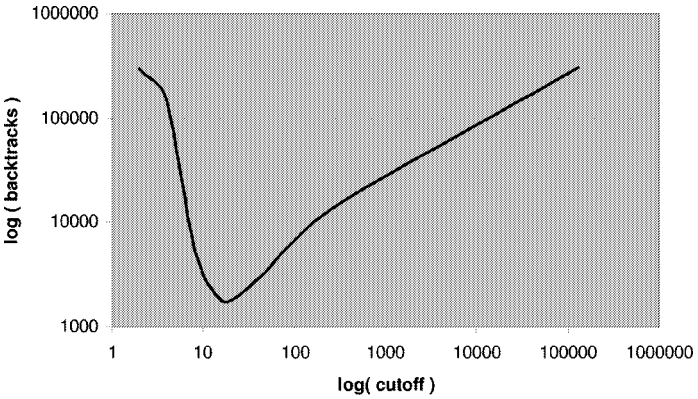


Figure 10. The effect of random restarts on solution cost for the logistics.d planning problem (SAT formulation).

4.3. RESTART RESULTS ON A RANGE OF PROBLEM INSTANCES

In Table IV we give the mean solution times of randomized backtrack procedures with rapid restarts, for a range of problem instances. The averages were computed over 100 runs each. For comparison, we included the runtime of the original deterministic procedures (Satz for the logistics, adder, and blocks-world instances; Ilog for the round-robin problems).*

Our deterministic Ilog procedure on the round-robin scheduling problem gives us a solution for the 14-team problem in 411 seconds; randomization improves this to 250 seconds. We could not find a solution for the 16- and 18-team problem with

* The deterministic runs can also be viewed as single runs of the randomized procedure with an infinite cutoff value. Note that, of course, one might be "lucky" on any any given instance and have the deterministic procedure branch in the right way. However, on the harder instances, we often need several hundred restarts to find a solution. On those instances, it becomes quite unlikely that a single deterministic run would succeed.

Table IV. Randomized rapid restarts (RRR) versus deterministic versions of backtrack search procedures (Satz solver used on SAT encodings; Ilog solver on CSP encodings).

Problem	Solver	Deterministic soln. time	RRR mean soln. time
logistics.d	Satz	108 min	95 sec
3bit-adder-32	Satz	$>$ 24 hrs	165 sec
3bit-adder-31	Satz	$>$ 24 hrs	17 min
round-robin 14	Ilog	411 sec	250 sec
round-robin 16	Ilog	$>$ 24 hrs	1.4 hrs
round-robin 18	Ilog	$>$ 48 hrs	$\approx$ 22 hrs
block-world.d	Satz	30 min	23 min

the deterministic version. Apparently, the subtle interaction between global and local constraints makes the search for a globally consistent solution surprisingly difficult. These problem instances are too hard to obtain a full cost distribution, making it difficult to calculate an optimal cutoff value.* For example, in the 16-team case, running with a cutoff of 1,000,000 gives a success rate of less than 40%. So, we do not even reach the median point of the distribution. Each run takes about 2 hours to complete. (We estimate that the median value is around 2,000,000.) In order to find a good cutoff value for very hard problem instances, one available strategy is a trial-and-error process, where one experiments with various cutoff values, starting at relatively low values, since the optimal cutoff for these problems tends to lie far below the median value of the distribution. Using such a strategy with RRR, we could solve the 16-team instance in 1.4 hours and the 18-team in approximately 22 hours.

For the SAT encodings of the 3-bit-adder problems (examples of Boolean circuit synthesis problems from the Dimacs benchmark [32]) the RRR solution times are – to the best of our knowledge – the first successful runs of a backtrack search procedure (Davis–Putnam–Logemann–Loveland) on these instances. (These instances were previously solved with local search methods [57].)

On the blocks-world problem, the table shows that we obtain little improvement over our deterministic result. We ran the randomized version of Satz on this instance over a wide range of cutoff values and with different levels of randomization ("heuristic equivalence" settings). However, there was no evidence of a heavy-tailed distribution, and, therefore, randomization only slightly increases the effectiveness of Satz.

* For problems for which we can empirically determine the overall cost profile, we can calculate the *optimal* cutoff value that minimizes expected cost of finding a solution.

The results in Table IV show that introducing a stochastic element into a backtrack-style SAT or CSP procedure, combined with an appropriate restart strategy, can significantly enhance the procedure's performance. In fact, as we see here, it allows us to solve several previously unsolved problem instances.

Niemela [49, 50], in independent work, made randomization and restarts a standard feature of his procedure for finding stable models of propositional theories. Niemela did not study directly the cost profiles of his search procedures, but empirically found that randomization and restarts can greatly boost the performance of his solver. Although we have not studied the cost profiles of Niemela's solver directly, it seems likely that they will be heavy tailed, which would explain the effectiveness of restarts.

Bayardo and Schrag [6], independently, included randomized tie-breaking and restarts in Relsat, but with only a fixed, high cutoff value. The focus of that work was on the effect of different backtrack techniques. The inclusion of random tie-breaking was based on the observation of the sensitivity of any fixed heuristics to individual problem instances [4].

Walsh [65] introduces a novel restart strategy in which the cutoff value increases geometrically. The advantage of such a strategy is that it is less sensitive to the details of the underlying heavy-tailed distribution. For further interesting related work, of a more purely theoretical nature, on minimizing the expected cost of randomized procedures and shortening the tails of randomized algorithms, see [2, 15, 40]. Luby et al. [40] provide a universal strategy for minimizing the expected cost of randomized procedures: one that is suitable for all distributions. For example, an efficient *universal* strategy consists of a sequence of runs whose lengths are powers of two, and each time a pair of runs of a given length has been completed, a run of twice that length is immediately executed. In our case, where we empirically determined the distribution of our search procedure, the optimal strategy is just a sequence of runs of a fixed length. Luby et al. [40] showed that such a fixed restart strategy is optimal when the underlying runtime distribution of the randomized procedure is fully known. The work behind Luby's universal strategies was motivated by Ertel's observation of possible long runs of theorem-proving methods and their effect on parallel strategies [15]. Although the universal strategy of Luby et al. is provable within a constant factor from optimal in the limit, we found that in practice the schedule often converges too slowly.

It should be noted that Luby et al.'s analysis for the universal strategy is based on two key assumptions: (1) no information about the prior distribution and (2) unlimited time resources. One can show that when one has some prior knowledge about the problem instance (which is reasonable if one has some experience with solving previous instances of a problem class) or when one has some time constraints (e.g., one needs to solve this instance within, say, two days of CPU time), then more sophisticated strategies are needed. Further work on selecting the best restart strategy is needed.

Other related work is that of Huberman et al. [30], who show how one can reduce expected performance on hard combinatorial problems by running several different heuristic algorithms in parallel in a "portfolio" approach. In [21], we show that portfolios are quite effective when mixing heavy-tailed distributions arising from different heuristic strategies.

5. Conclusions

Heavy-tailed probability distributions have been used to model a variety of real-world phenomena, such as weather patterns and delays in communication networks. Our results show the suitability of such distributions in modeling the runtime behavior of Davis–Putnam style satisfiability procedures, which are the current fastest complete SAT procedures. Such procedures are based on a backtrack search through the space of possible truth assignments. Our results elucidate a series of puzzling characteristics of the computational cost of Davis–Putnam style methods. In particular, the heavy-tail model explains the informally observed highly erratic behavior of the mean and variance, as well as the long tails of the cost profiles. After observing such distributions in a variety of practical domains (including CSP formulations), it appears that the heavy-tailed behavior may be the rule rather than the exception for backtrack search methods.

The understanding of these characteristics explains why a "rapid restart" strategy is an effective remedy against heavy-tailed phenomena. In addition, restarts can exploit the mass of probability on the left of the cost distributions. Restarts therefore reduce the variance in runtime and the probability of failure of the search procedure, resulting in a more robust overall search method.

An interesting question for future research is to determine what characteristics of SAT instances lead to heavy-tailed behavior and exactly how the heavy-tailedness depends on the backtrack search strategy itself. For dealing with unsolved problem instances, one would want to develop measures that are predictive of heavy-tailedness for a problem instance without having to obtain cost profiles of actual runs on the instance. If such measures can be found, one could develop a priori optimal or near-optimal restart schedules. However, even without such measures, we have found that one can often use restarts effectively because good schedules tend to be relatively uniform within a problem domain.

Finally, as we discussed, there are still many interesting research questions regarding the best restart policy. However, in practical terms [4, 38], we already see that randomization and restarts are being added to the state-of-the-art complete satisfiability procedures.

Acknowledgments

We thank Ian Gent, Toby Walsh, and the anonymous reviewers for useful comments and feedback. We also thank Karen Alguire, Roberto Bayardo, Nort Fowler,

Neal Glassman, Joseph Halpern, Holger Hoos, Richard Karp, Jean Charles Regin, Gennady Samorodnitsky, and Mark Stickel for useful suggestions and discussions.

The first author is supported by the Air Force Research Laboratory and the Air Force Office of Scientific Research, under the New World Vistas Initiative. The second author is supported by an NSF Faculty Early Career Development Award, an Alfred P. Sloan fellowship, and by the Air Force Research Laboratory. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. government.

Appendix

A. Adaptation of Hill Estimator

Assume a sample of $k = n + u$ i.i.d. random variables is drawn. Let the order statistics for the smallest n ($n \leq k$) values on the sample be $X_{n1} \leq X_{n2} \leq \cdots \leq X_{nn}$. Assume that, for $X_{n,n-r} \leq X \leq X_{nn}$, the tail distribution is of the Paul Lévy type. We will show that the conditional maximum likelihood estimator for the maximal moment exponent α is

$$\hat{\alpha}_{r,u} = \left(\frac{1}{r} \sum_{j=1}^{r-1} \ln X_{n,n-r+j} + \frac{u+1}{r} \ln X_{n,n} - \frac{u+r}{r} \ln X_{n,n-r} \right)^{-1} \tag{4}$$

and its variance is given by

$$\widehat{\mathrm{Var}}(\hat{\alpha}) = \frac{\hat{\alpha}^2 (r+1)^2}{r^2 (r-1)}.$$

In order to simplify the derivation, we will change momentarily the notation and derive the estimator in the context of a lower tail of the Pareto–Lévy form. The modification of the estimator for dealing with upper-tail values is straightforward. We will follow the approach of Hill (1975). For further details, see [10].

Let the order statistics be

$$Z^{(k)} \leq Z^{(k-1)} \leq \cdots \leq Z^{(l+s)} \leq \cdots \leq Z^{(l+1)} \leq Z^{(l)} \leq \cdots \leq Z^{(2)} \leq Z^{(1)}.$$

We have a total of k random variables. We assume that for $Z \leq Z^{(l)}$, at least, these are of the Pareto–Lévy type. However, we observe only the $l + s + 1$ variables $Z^{(l+s)} \leq \cdots \leq Z^{(l+1)} \leq Z^{(l)}$, and not the $k - l - s$ variables $Z^{(k)} \leq \cdots \leq Z^{(l+s+1)}$. Thus, only the $s + 1$ variables $Z^{(l+s)} \leq Z^{(l+s-1)} \leq \cdots \leq Z^{(l)}$ are included in the maximum likelihood estimator.

As in Hill (1975), the derivation is based on the Renyi (1953) representation theorem. Assume that the cumulative distribution function for Z, F, is continuous and strictly increasing. Then,

$$Z^{(i)} = F^{-1}\left(\exp\left\{ -\frac{e_1}{k} - \frac{e_2}{k-1} - \cdots - \frac{e_i}{k-i+1} \right\} \right),$$

where the e_i are independent exponential random variables with mean 1. Restricting our attention to the $s+1$ variables $Z^{(l+s)} \leq \cdots \leq Z^{(l+1)} \leq Z^{(l)}$, we have

$$\ln F(Z^{(l+i)}) = -\frac{e_1}{k} - \frac{e_2}{k-1} - \cdots - \frac{e_{l+i}}{k-l-i+1}$$

and so

$$(k-l-i+1)\big(\ln F(Z^{(l+i)}) - \ln F(Z^{(l+i+1)})\big) = e_i, \quad \text{for } i = 0, 1, \ldots, s.$$

Unlike Hill (1975) we assume directly that $F(z) = Cz^{\alpha}$ for the range above, which rewritten as $\ln F(z) = C + \alpha \ln z$ and introduced in the previous equation shows that the random variables T_j

$$T_{k-l-i+1} = (k-l-i+1)\big(\ln Z^{(l+i)} - \ln Z^{(l+i+1)}\big) = e_i/\alpha, \quad \text{for } i = 0, 1, \ldots, s,$$

are independent and exponentially distributed with parameter α. Therefore, conditioning on the boundary variables $Z^{(l)}$ and $Z^{(l+s)}$, the maximum likelihood estimator for α is $\hat{\alpha} = s/\sum T_i$. It is easy to verify that the sum $\sum T_i$ collapses to

$$(k-l)\ln Z^{(l)} - \sum_{i=1}^{s-1} \ln Z^{(l+i)} - (k-l-s+1)\ln Z^{(l+s)}.$$

Therefore, the estimator becomes

$$\hat{\alpha} = \left(\frac{k-l}{s}\ln Z^{(l)} - \frac{1}{s}\sum_{i=1}^{s-1}\ln Z^{(l+i)} - \frac{k-l-s+1}{s}\ln Z^{(l+s)}\right)^{-1}.$$

To modify this estimator for working with upper tails of the Pareto–Lévy type, we just note that letting $Z = X^{-1}$ we have $\mathrm{P}\{Z \leq x\} = \mathrm{P}\{X \geq x^{-1}\} = Cx^{-\alpha}$. The changes lead directly to Equation (4).

An estimator of the estimator's variance can be obtained by noticing that $\hat{\alpha} = s/\sum T_i$ has an inverted gamma distribution. Therefore

$$\widehat{\mathrm{Var}}(\hat{\alpha}) = \frac{\hat{\alpha}^2(r+1)^2}{r^2(r-1)}.$$

B. A Formal Characterization of Restarts

In this section, we formalize the restart strategy of a complete randomized backtrack search method. We show that the probability distribution associated with such a restart strategy does not exhibit heavy tails. Furthermore, the moments of the restart strategy are finite.

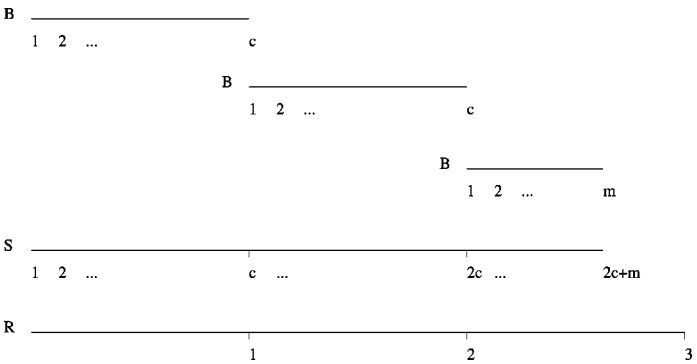


Figure 11. B – number of choice points searched by the randomized backtrack procedure (running with cutoff c); S – number of choice points searched by RRR; R – number of runs executed by RRR. In this case, a solution was found in the third run ($R = 3$), with a total of $2c + m$ choice points ($S = 2c + m$). The third run of the randomized backtrack search method took $m \leq c$ choice points.

Given a randomized backtrack search procedure, we consider the number of choice points (or backtracks) performed by such a procedure. We introduce random variable B such that

> B is the number of choice points that the backtrack search procedure takes to find a solution or prove that it does not exist. $B = \{1, 2, \ldots\}$.

Now consider a Rapid Randomized Restarts (RRR) strategy for running our backtrack procedure: run the procedure up to a fixed number of choice points c (the cutoff); if the procedure finds a solution or proves that no solution exists, then RRR has also found a solution (or proven that no solution exists) and stops; otherwise restart the backtrack procedure from the beginning (using an independent random seed) for another c decision events, and so on. We associate with RRR the random variable S such that

> S is the number of choice points that RRR takes to find a solution or prove that no solution exists. $S = \{1, 2, \ldots\}$.

Let's define a "run" as the execution of the randomized backtrack search method for up to c steps. We now define the random variable R, such that

> R is the number of runs executed by RRR.

Figure 11 illustrates how the different random variables relate to each other.

The runs executed by RRR are independent (no information is carried over between runs, and each run uses a new random seed) and therefore can be seen as a sequence of Bernoulli trials, in which the success of a trial corresponds to finding a solution (or proving that one does not exist) during a run; its probability is given by P[B$\leq c$]. Therefore, R follows a *geometric distribution* with parameter $p = P[B \leq c]$.

The probability of the tail of S, $P[S > s]$, corresponds to the probability of not finding the solution in the first $\lfloor s/c \rfloor$ runs of RRR, and finding it with more than ($s \bmod c$) choice points in the next run. We obtain the following expression:

$$P[S > s] = P[B > c]^{\lfloor s/c \rfloor} \, P[B > s \bmod c]. \tag{5}$$

The distribution of S is not heavy tailed because its tail exhibits exponential decay:

$$P[S > s] \leq P[B > c]^{\lfloor s/c \rfloor} = P[R > \lfloor s/c \rfloor]. \tag{6}$$

In words, the tail of S is limited from above by the tail of R. Since R follows a geometric distribution, it has finite moments, and therefore so does S. The full distribution of S is given by the following expression:

$$P[S = s] = \begin{cases} P[B > c]^{\lfloor s/c \rfloor} \, P[B = s \bmod c], & s \bmod c \neq 0 \\ P[B > c]^{\lfloor s/c \rfloor - 1} \, P[s = c], & \text{otherwise.} \end{cases} \tag{7}$$

Note that the second branch of (7) corresponds to the case in which the total number of choice points executed by strategy S is a multiple of c. This situation occurs when the solution is found when the cutoff c is reached.

Based on the distribution of B, we can determine a cutoff, c, that minimizes the expected runtime of S. Alternatively, one can determine a cutoff value that minimizes both the expected runtime and variance of the RRR strategy, using the tools from portfolio analysis (e.g., [30, 21]). In our experiments, we determined the cutoff for the restart strategy (RRR) based on the empirical distribution of B, which was computed, when possible, by performing $10,000$ runs of the backtrack search methods with a very high cutoff.

References

1. Adler, R., Feldman, R. and Taggu, M. (eds.): *A Practical Guide to Heavy Tails*, Birkhäuser, 1998
2. Alt, H., Guibas, L., Mehlhorn, K., Karp, R. and Wigderson, A.: A method for obtaining randomized algorithms with small tail probabilities, *Algorithmica* **16**(4–5) (1996), 543–547.
3. Anderson, L.: Completing partial latin squares, *Math. Fys. Meddelelser* **41** (1985), 23–69.
4. Bayardo, R.: Personal communications, 1999.
5. Bayardo, R. and Miranker, D.: A complexity analysis of space-bounded learning algorithms for the constraint satisfaction problem, in *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, Portland, OR, 1996, pp. 558–562.
6. Bayardo, R. and Schrag, R.: Using CSP look-back techniques to solve real-world SAT instances, in *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, New Providence, RI, 1997, pp. 203–208.
7. Brelaz, D.: New methods to color the vertices of a graph, *Comm. ACM* **22**(4) (1979), 251–256.
8. Colbourn, C.: Embedding partial Steiner triple systems is NP-complete, *J. Combin. Theory A* **35** (1983), 100–105.
9. Cook, S. and Mitchell, D.: Finding hard instances of the satisfiability problem: A survey, in Du, Gu, and Pardalos (eds), *Satisfiability Problem: Theory and Applications*, Dimacs Series in Discrete Math. and Theoret. Comput. Sci. 35, 1998.

10. Crato, N., Gomes, C. and Selman, B.: On estimating the index of stability with truncated data, Manuscript in preparation.
11. Crawford, J. and Baker, A.: Experimental results on the application of satisfiability algorithms to scheduling problems, in *Proceedings of The Tenth National Conference on Artificial Intelligence (AAAI-94)*, Austin, TX, 1994, pp. 1092–1098.
12. Davis, M., Logemann, G. and Loveland, D.: A machine program for theorem proving, *Comm. ACM* **5** (1979), 394–397.
13. Davis, M. and Putman, H.: A computing procedure for quantification theory, *J. ACM* **7** (1960), 201–215.
14. Denes, J. and Keedwell, A.: *Latin Squares and Their Applications*, Akademiai Kiado, Budapest, and English Universities Press, 1974.
15. Ertel, W. and Luby, M.: Optimal parallelization of Las Vegas algorithms, in *Symp. on Theoretical Aspects of Computer Science*, Lecture Notes in Comput. Sci. 775, 1994, pp. 463–475.
16. Feller, W.: *An Introduction to Probability Theory and Its Applications*, Vol. I, Wiley, New York, 1968.
17. Feller, W.: *An Introduction to Probability Theory and Its Applications*, Vol. II, Wiley, New York, 1971.
18. Frost, D., Rish, I. and Vila, L.: Summarizing CSP hardness with continuous probability distributions, in *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, New Providence, RI, 1997, pp. 327–334.
19. Fujita, M., Slaney, J. and Bennett, F.: Automatic generation of some results in finite algebra, in *Proceedings of the International Joint Conference on Artificial Intelligence*, France, 1993, pp. 52–57.
20. Gent, I. and Walsh, T.: Easy problems are sometimes hard, *Artif. Intell.* **70** (1993), 335–345.
21. Gomes, C. P. and Selman, B.: Algorit portfolio design: Theory vs. practice, in *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI-97)*, Providence, RI, 1997, pp. 190–197.
22. Gomes, C. P. and Selman, B.: Problem structure in the presence of perturbations, in *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, New Providence, RI, 1997, pp. 221–227.
23. Gomes, C. P., Selman, B. and Crato, N.: Heavy-tailed distributions in combinatorial search, in G. Smolka (ed.), *Principles and Practice of Constraint Programming (CP97)*, Lecture Notes in Comput. Sci., Linz, Austria, 1997, pp. 121–135.
24. Hall, P.: On some simple estimates of an exponent of regular variation, *J. Roy. Statist. Soc.* **44** (1982), 37–42.
25. Hayes, B.: Computer science: Can't get no satisfaction, *American Scientist* **85**(2) (1996), 108–112.
26. Hill, B.: A simple general approach to inference about the tail of a distribution, *Ann. Statist.* **3** (1975), 1163–1174.
27. Hogg, T., Huberman, B. and Williams, C. (eds.): Phase transitions and complexity (Special Issue), *Artif. Intell.* **81**(1–2) (1996).
28. Hogg, T. and Williams, C.: Expected gains from parallelizing constraint solving for hard problems, in *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, Seattle, WA, 1994, pp. 1310–1315.
29. Hoos, H.: Stochastic local search – methods, models, applications, Ph.D. Thesis, TU Darmstadt, 1998.
30. Huberman, B., Lukose, R. and Hogg, T.: An economics approach to hard computational problems, *Science* **275** (1993), 51–54.
31. Johnson, D.: A theoretician's guide to the experimental analysis of algorithms, Preliminary manuscript. Invited talk presented at AAAI-96, Portland, OR, 1996.

32. Johnson, D. and Trick, M.: Cliques, coloring, and satisfiability: Second dimacs implementation challenge, in Dimacs Series in Discrete Math. and Theoret. Comput. Sci. 36, 1996.
33. Kamath, A., Karmarkar, N., Ramakrishnan, K. and Resende, M.: Computational experience with an interior point algorithm on the satisfiability problem, in *Proceedings of Integer Programming and Combinatorial Optimization*, Waterloo, Canada, 1990, pp. 333–349.
34. Kautz, H. and Selman, B.: Pushing the envelope: Planning, propositional logic, and stochastic search, in *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, Portland, OR, 1996, pp. 1188–1194.
35. Kirkpatrick, S. and Selman, B.: Critical behavior in the satisfiability of random Boolean expressions, *Science* **264** (1994), 1297–1301.
36. Kwan, A.: Validity of normality assumption in CSP research, in *Proceedings of the Pacific Rim International Conference on Artificial Intelligence*, 1995, pp. 459–465.
37. Lam, C., Thiel, L. and Swiercz, S.: The non-existence of finite projective planes of order 10, *Canad. J. Math.* **XLI**(6) (1994), 1117–1123.
38. Li, C. M.: Personal communications, 1999.
39. Li, C. M. and Anbulagan: Heuristics based on unit propagation for satisfiability problems, in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1997, pp. 366–371.
40. Luby, M., Sinclair, A. and Zuckerman, D.: Optimal speedup of Las Vegas algorithms, *Inform. Process. Lett.*, 1993, pp. 173–180.
41. Macready, W., Siapas, A. and Kauffman, S.: Criticality and parallelism in combinatorial optimization, *Science* **271** (1996), 56–59.
42. Mandelbrot, B.: The Pareto–Lévy law and the distribution of income, *Internat. Econom. Rev.* **1** (1960), 79–106.
43. Mandelbrot, B.: The variation of certain speculative prices, *J. Business* **36** (1963), 394–419.
44. Mandelbrot, B.: *The Fractal Geometry of Nature*, Freeman, New York, 1983
45. McAloon, K., Tretkoff, C. and Wetzel, G.: Sports league scheduling, in *Proceedings of Third Ilog International Users Meeting*, 1997.
46. Mitchell, D. and Levesque, H.: Some pitfalls for experimenters with random SAT, *Artif. Intell.* **81**(1–2) (1996), 111–125.
47. Monasson, R., Zecchina, R., Kirkpatrick, S., Selman, B. and Troyansky, L.: Determining computational complexity from characteristic 'phase transitions', *Nature* **400**(8) (1999), 133–137.
48. Nemhauser, G. and Trick, M.: Scheduling a major college basketball conference, *Oper. Res.* **46**(1) (1998), 1–8.
49. Niemela, I.: Logic programs with stable model semantics as a constraint programming paradigm. Ext. version of paper presented at the *Workshop on Computational Aspects of Nonmonotonic Reasoning*, 1998.
50. Niemela, I.: Personal communications, 1999.
51. Nolan, J.: Stable distributions, Manuscript, in preparation, 1999.
52. Prosser, P.: Hybrid algorithms for the constraint satisfaction problem, *Comput. Intell.* **9**(3) (1993), 268–299.
53. Puget, J. F. and Leconte, M.: Beyond the black box: Constraints as objects, in *Proceedings of ILPS'95*, Lisbon, Portugal, 1995, pp. 513–527.
54. Regin, J. C.: A filtering algorithm for constraints of difference in CSP, in *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, Seattle, WA, 1994, pp. 362–367.
55. Rish, I. and Frost, D.: Statistical analysis of backtracking on inconsistent CSPs, in *Proceedings of Third International Conference on Principles and Practices of Constraint Programming*, 1997, pp. 150–162.
56. Samorodnitsky, G. and Taqqu, M.: *Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance*, Chapman and Hall, 1994.

57. Selman, B., Kautz, H. and Cohen, B.: Local search strategies for satisfiability testing, in D. Johnson and M. Trick (eds), Dimacs Series in Discrete Math. and Theoret. Comput. Sci. 26, 1993, pp. 521–532.
58. Selman, B. and Kirkpatrick, S.: Finite-size scaling of the computational cost of systematic search, *Artif. Intell.* **81**(1–2) (1996), 273–295.
59. Shaw, P., Stergiou, K. and Walsh, T.: Arc consistency and quasigroup completion, in *Proceedings of ECAI-98, Workshop on Binary Constraints*, 1998.
60. Shonkwiler, R., E. and van Vleck, E.: Parallel speedup of Monte Carlo methods for global optimization, *J. Complexity* **10** (1994), 64–95.
61. Slaney, J., Fujita, M. and Stickel, M.: Automated reasoning and exhaustive search: Quasigroup existence problems, *Comput. Math. Appl.* **29** (1995), 115–132.
62. Smith, B. and Grant, S.: Sparse constraint graphs and exceptionally hard problems, in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1995, pp. 646–651.
63. Stickel, M.: Personal communications, 1998.
64. Vandengriend and Culberson: The G_{nm} phase transition is not hard for the Hamiltonian cycle problem, *J. Artif. Intell. Res.* **9** (1999), 219–245.
65. Walsh, T.: Search in a small world, in *Proceedings of the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 1999
66. Zolotarev, V.: *One-Dimensional Stable Distributions*, Transl. Math. Monographs 65, Amer. Math. Soc., 1986. Translation from the original 1983 Russian ed.