Summarizing CSP hardness with continuous probability distributions

Daniel Frost, Irina Rish, and Lluís Vila
Dept. of Information and Computer Science
University of California, Irvine, CA 92717-3425
{dfrost,irinar,vila}@ics.uci.edu

Abstract

We present empirical evidence that the distribution of effort required to solve CSPs randomly generated at the 50% satisfiable point, when using a backtracking algorithm, can be approximated by two standard families of continuous probability distribution functions. Solvable problems can be modelled by the Weibull distribution, and unsolvable problems by the lognormal distribution. These distributions fit equally well over a variety of backtracking based algorithms.

1. Introduction

Several key developments in the 1990's have contributed to the advancement of empirical research on CSP algorithms, to the extent that the field may even be called an experimental science. Striking increases in computer power and decreases in cost, coupled with the general adoption of C as the programming language of choice, have made it possible for the developer of a new algorithm or heuristic to test it on large numbers of random instances. Another important advance was the recognition of the "50% satisfiable" phenomenon (Mitchell, Selman, & Levesque 1992), which has enabled researchers to focus on the hardest problems.

It is often not clear which measures to report from large scale experiments. The usual parameter of interest is the cost of solving a problem, measured by CPU time, number of consistency checks, or size of search space. The mean and the median are the most popular statistics, but these do not capture the long "tail" of difficult problems that often occurs. In order to convey more information, some authors have reported percentile points such as the hardness of the problem at the 99th and 99.9th percentiles, minimum and maximum values, and the standard deviation. To illustrate the problem, consider an experiment with 200 CSP instances, 198 requiring between .5 and 10 seconds to solve, one 25 seconds, and one 100 seconds. How can these results be clearly and concisely reported? In experiments involving a large set of randomly generated instances, the ideal would be to report the entire distribution of cost to solve.

In this paper we present empirical evidence that the distribution of the number of consistency checks required to solve randomly generated CSPs, when generated at the 50% satisfiable point and using backtracking based algorithms, can be approximated by two standard families of continuous probability distribution functions. Solvable problems are modelled reasonably well by the Weibull distribution. The lognormal distribution fits the unsolvable problems with a high degree of statistical significance. Each of these distributions is actually a family of distributions, with specific functions characterized by a scale parameter and a shape parameter. We measure the goodness-of-fit of our results using the chi-square statistic.

By noting that the results of an experiment can be fit by distribution D with parameters x and y, it is possible to convey a complete understanding of the experimental results: the mean, median, mode, and shape of the tail. If the distribution of hardness is known to be quite similar to a continuous distribution, several other benefits may accrue. Experimenting with a relatively small number of instances can permit the shape and scale parameters of the distribution to be estimated. Well-developed statistical techniques, based on the assumption of a known underlying distribution, are available for estimating parameters based on data that have been "censored" above a certain point (Nelson 1990). This may aid the interpretation of an experiment in which runs are terminated after a certain time point. Knowing the distribution will also enable a more precise comparison of competing algorithms. For instance, it is easier to determine whether the difference in the means of two experiments is statistically significant if the population distributions are known.

Finally, we believe that pursuing the line of inquiry we initiate here will lead to a better understanding of both random problem generators and backtracking based search. The Weibull and lognormal distributions have interpretations in engineering and the sciences

 This work was partially supported by NSF grant IRI-9157636 and by Air Force Office of Scientific Research grant AFOSR 900136.

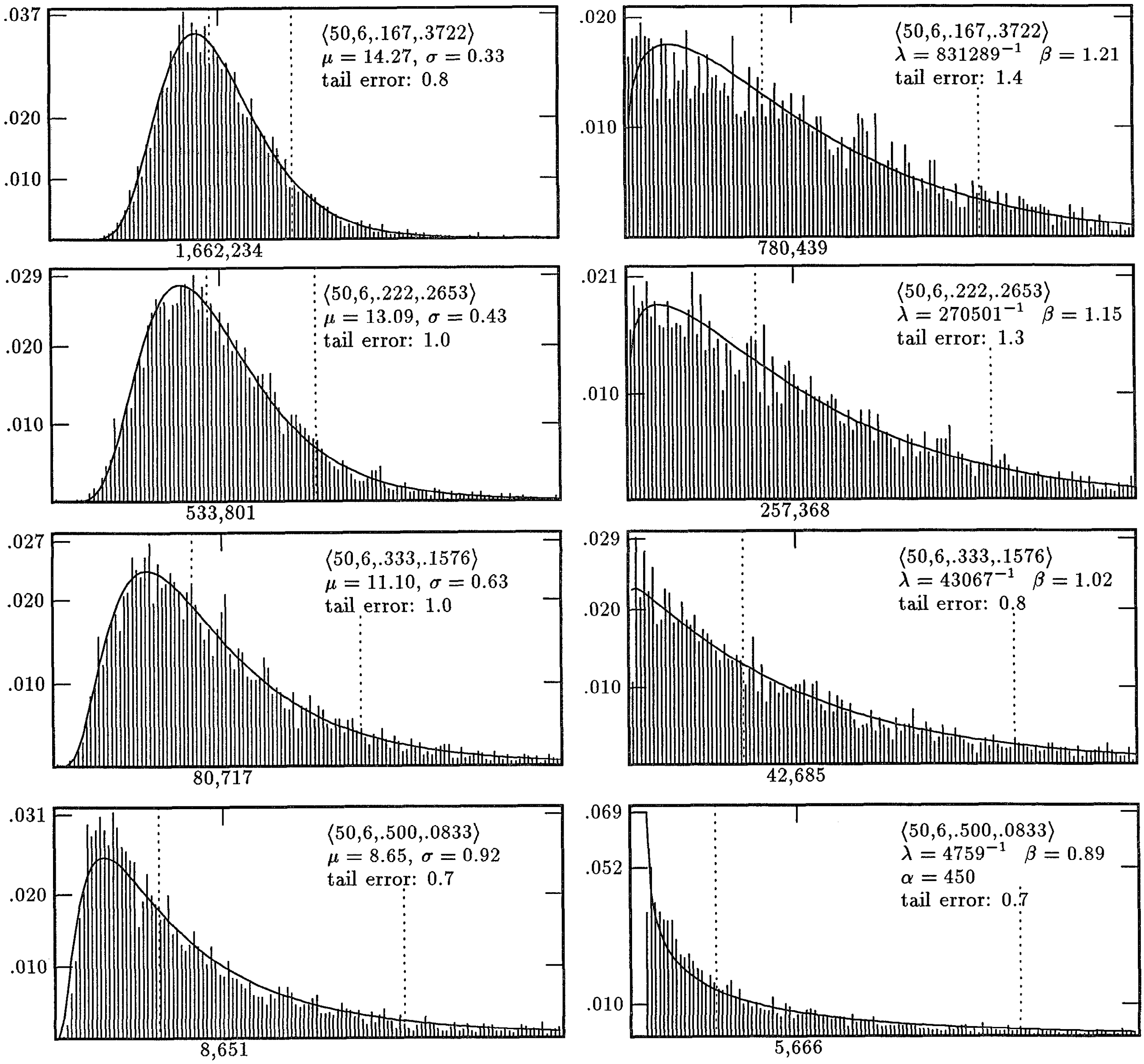

Figure 1: Graphs of sample data (vertical bars) and continuous distributions (curved lines) for selected experiments, using algorithm BJ+DVO. Unsolvable problems and lognormal distributions are shown on the left; solvable problems and Weibull distributions on the right. Note that the scales vary from graph to graph, and the right tails have been truncated. The x-axis unit is consistency checks; the sample mean is indicated. The data has been grouped in ranges equal to one fortieth of the mean. The y-axis shows the fraction of the sample that is expected (for the distribution functions) or was found to occur (for the experimental data) within each range of consistency checks. The vertical dotted lines indicate the median and 90th percentile of the data.

which may provide insight into the search process.

2. Problems and Algorithms

The constraint satisfaction problem (Dechter 1992) is used to model many areas of interest to Artificial Intelligence. A CSP has a set of variables, and solving the problem requires assigning to each variable a value from a specified finite domain, subject to a set of constraints which indicate that when certain variables have certain values, other variables are prohibited from being assigned particular values. In this paper, we consider the task of looking for a single assignment of values to variables that is consistent with all constraints, or for a proof that no such assignment exists.

We experiment with several standard algorithms from the literature, and here give references and abbreviations. The base algorithm is simple chronological backtracking (BT) (Bitner & Reingold 1975) with no variable or value ordering heuristic (a fixed random ordering is selected before search). We also use conflict-directed backjumping (BJ) (Prosser 1993) with no ordering heuristic, backtracking with the min-width variable ordering heuristic (BT+MW) (Freuder 1982), and forward checking (FC) (Haralick & Elliott 1980) with no variable ordering heuristic. As an example of a more sophisticated algorithm that combines backjumping, forward checking style domain filtering, and a dynamic variable ordering scheme, we use BJ+DVO from (Frost & Dechter 1994). For the 3SAT problems, we use the Davis-Putnam procedure (DP) (Davis, Logemann, & Loveland 1962) with no variable ordering heuristic, and augmented with a set of sophisticated ordering heuristics (DP+HEU) (Crawford & Auton 1993).

The binary CSP experiments reported in this paper were run on a model of uniform random binary constraint satisfaction problems that takes four parameters: N, D, T and C. The problem instances are binary CSPs with N variables, each having a domain of size D. The parameter T (tightness) specifies a fraction of the D^2 value pairs in each constraint that are disallowed by the constraint. The value pairs to be disallowed by the constraint are selected randomly from a uniform distribution, but each constraint has the same fraction T of such incompatible pairs. The parameter C specifies the proportion of constraints out of the $N*(N-1)/2$ possible; C ranges from 0 (no constraints) to 1 (a complete graph). The specific constraints are chosen randomly from a uniform distribution. We specify the parameters between angle brackets: $\langle N, D, T, C\rangle$. This model is the binary CSP analog of the Random KSAT model described in (Mitchell, Selman, & Levesque 1992), and has been widely used by many researchers (Prosser 1996; Smith & Dyer 1996). We also report some experiments with 3SAT problems, which can be viewed as a type of CSP with ternary constraints and $D = 2$.

All experiments reported in this paper were run with parameters that produce problems in the 50% satisfiable region. These combinations were determined empirically.

3. Continuous probability distributions

In this section we briefly describe two probability distributions well known in the statistics literature. Each is characterized by a cumulative distribution function (cdf), which for a random variable T is defined to be

$$F(t) = P(T \leq t), -\infty < T < \infty$$

and a probability density function $f(t) = dF(t)/dt$.

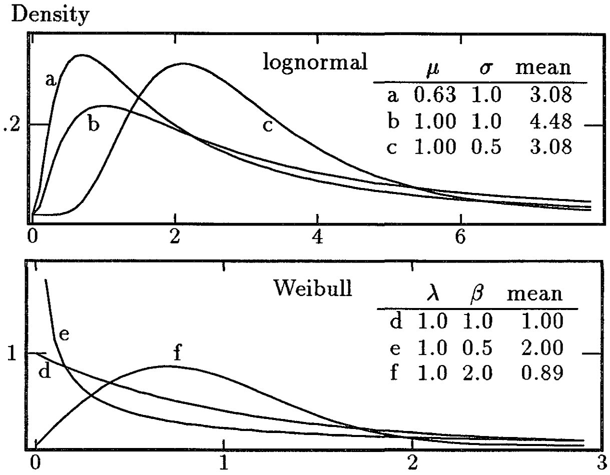

Figure 2: Graphs of the lognormal and Weibull density functions for selected parameter values.

Weibull. The Weibull distribution uses a scale parameter λ and a shape parameter β. Its density function is

$$f(t) = \begin{cases} \lambda^\beta \beta t^{\beta-1} e^{-(\lambda t)^\beta} & t > 0 \\ 0, & t \leq 0 \end{cases}$$

and the cdf is

$$F(t) = \begin{cases} 1 - e^{-(\lambda t)^\beta}, & t > 0 \\ 0, & t \leq 0. \end{cases}$$

The mean, E, of a Weibull distribution is given by $E = \lambda^{-1}\Gamma(1+\beta^{-1})$ where $\Gamma(\cdot)$ is the Gamma function. There is also a three parameter version of the Weibull distribution, in which t is replaced by $t-\alpha$ in the above equations; α is called the *origin* of the distribution. We use the three-parameter version when the mean of our sample is small, e.g. with $\langle 50, 6, .500, .0833\rangle$ and BJ+DVO (see Fig. 1). When $\beta = 1$, the Weibull distribution is identical to the exponential distribution.

Lognormal. The lognormal distribution is based on the well-known normal or Gaussian distribution. If the logarithm of a random variable is normally distributed, then the random variable itself shows a lognormal distribution. The density function, with scale parameter μ and shape parameter σ, is

$$f(t) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma t} \exp\left(\frac{-(\log t - \mu)^2}{2\sigma^2}\right), & t > 0 \\ 0, & t \leq 0 \end{cases}$$

and the lognormal distribution function is

$$F(t) = \Phi\left(\frac{\log t - \mu}{\sigma}\right),$$

where $\Phi(\cdot)$ is the normal cumulative distribution function. The mean value of the lognormal distribution is $E = e^{\mu+\sigma^2/2}$. Simple formulas for the median and mode are given by e^{μ} and $e^{\mu-\sigma^2}$, respectively. See Fig. 2 for the forms of the Weibull and lognormal density functions.

Estimating Parameters

Given a population sample and a parameterized probability distribution family, there are several methods for estimating the parameters that best match the data. For the Weibull distribution we employed the maximum likelihood estimator described in (D'Agostino & Stephens 1986). A simple maximum likelihood estimator for the lognormal distribution is described in (Aitchison & Brown 1957), but we found that this approach produced parameters that fit the far right tail extremely accurately, but did not match the data overall, as evidenced by both visual inspection and the chi-square statistic described below. Therefore, we report parameters for the lognormal distribution based on minimizing a "homegrown" error function. This function groups the sorted data into ten intervals, each with the same number of instances. Let s_i and e_i be the endpoints of interval i, $s_1 = 0$, $e_i = s_{i+1}$, and $e_{10} = \infty$. Define $R_i = (F(e_i) - F(s_i))/0.1$, where F is the cdf of the distribution. If $R_i < 1$, then $R_i \leftarrow 1/R_i$. The error function is then ΣR_i.

We note that for both distributions, the parameters are computed so that the mean of the distribution is identical to the sample mean.

Statistical Significance and Tail Error Test

To measure goodness-of-fit we frame a "null hypothesis" that the random sample of data from our experiment is from the distribution $F_0(x)$ (either lognormal or Weibull). To test this hypothesis, we order the samples by ascending number of consistency checks and partition them into M bins. If o_i is the number of instances in the ith bin as observed in the experiment, and e_i is the expected number in the bin according to the distribution (with specific parameters), then Pearson's chi-square statistic is

$$\chi^2 = \sum_{i=1}^{M} \frac{(o_i - e_i)^2}{e_i}$$

To interpret this statistic we need to know ν, the number of degrees of freedom. ν is computed by taking the number of bins and subtracting one plus the number of parameters that have been estimated from the data. Thus $\nu = M - 3$. We compute M following a recommendation in (D'Agostino & Stephens 1986): $M = 2m^{2/5}$, where m is the sample size.

Knowing χ^2 and ν, we can ask if the evidence tends to support or refute the null hypothesis. By referencing a table or computing the chi-square probability function with χ^2 and ν (as we have done), we determine the significance level at which we can accept the null hypothesis. The higher the level of significance, the more the evidence tends not to refute the null hypothesis. In Fig. 4, we indicate which sets of data were fit by the lognormal distribution at the .95 significance level by printing the μ and σ parameters in bold type.

The chi-square test gives equal importance to goodness-of-fit over the entire distribution, but sometimes experimenters are particularly interested in the behavior of the rare hard problems in the right tail. Therefore, we have devised a simple measure of "tail error." To compute the tail error measure, we find the number of consistency checks for the instance at the 99th percentile. For example, out of 5,000 instances the 4,950th hardest one might have needed 2,000,000 consistency checks. We then plug this number into the cdf: $x = F(2{,}000{,}000)$. The tail error measure is $(1.0 - x)/(1.0 - .99)$, where x is the probability of an instance being less than 2,000,000 according to the distribution, and .99 is the fraction of the data that was less. If the result is 1.0, the match is perfect. A number less than 1 indicates that the distribution does not predict as many instances harder than the 99th percentile instance as were actually encountered; when greater than 1 the measure indicates the distribution predicts too many such hard problems.

4. Experimental procedure

Our experimental procedure consisted of selecting various sets of parameters for the random CSP generator, generating 10,000 instances for each set, and selecting an algorithm to use. For each instance we recorded whether a solution was found and the number of consistency checks required to process it. Employing the estimators refered to above, we derived parameters for the Weibull and lognormal distributions, and measured the statistical significance of the fit using the χ^2 statistic. Each line in Fig. 4 represents one experiment with one algorithm on the unsolvable instances from one set of parameters. We report extensively on unsolvable instances only, since only for those problems did we find a statistically significant fit to a continuous probability distributions. Some of our experimental results are shown graphically in Fig. 1 and Fig. 3.

The column labeled "Mean" in Fig. 4 shows the mean number of consistency checks for the experiment, rounded to the nearest thousand and final 000 truncated. The "μ" and "σ" columns show the computed value for these parameters, in bold when the fit is statistically significant at the .95 level. The fit was significant at the .90 level otherwise. The tail error measure is reported in the "Tail" column.

Setting N=20 and D=6, we experimented with four combinations of T and C, and four different algorithms, BT, BJ, FC, and BT+MW. We selected a variety of relatively simple algorithms in order to demonstrate that the correspondence with continuous distributions is not the consequence of any specific heuristic, but holds for many varieties of backtracking search. The range of values for T and C show that the distributions fit the data over a range of graph density and

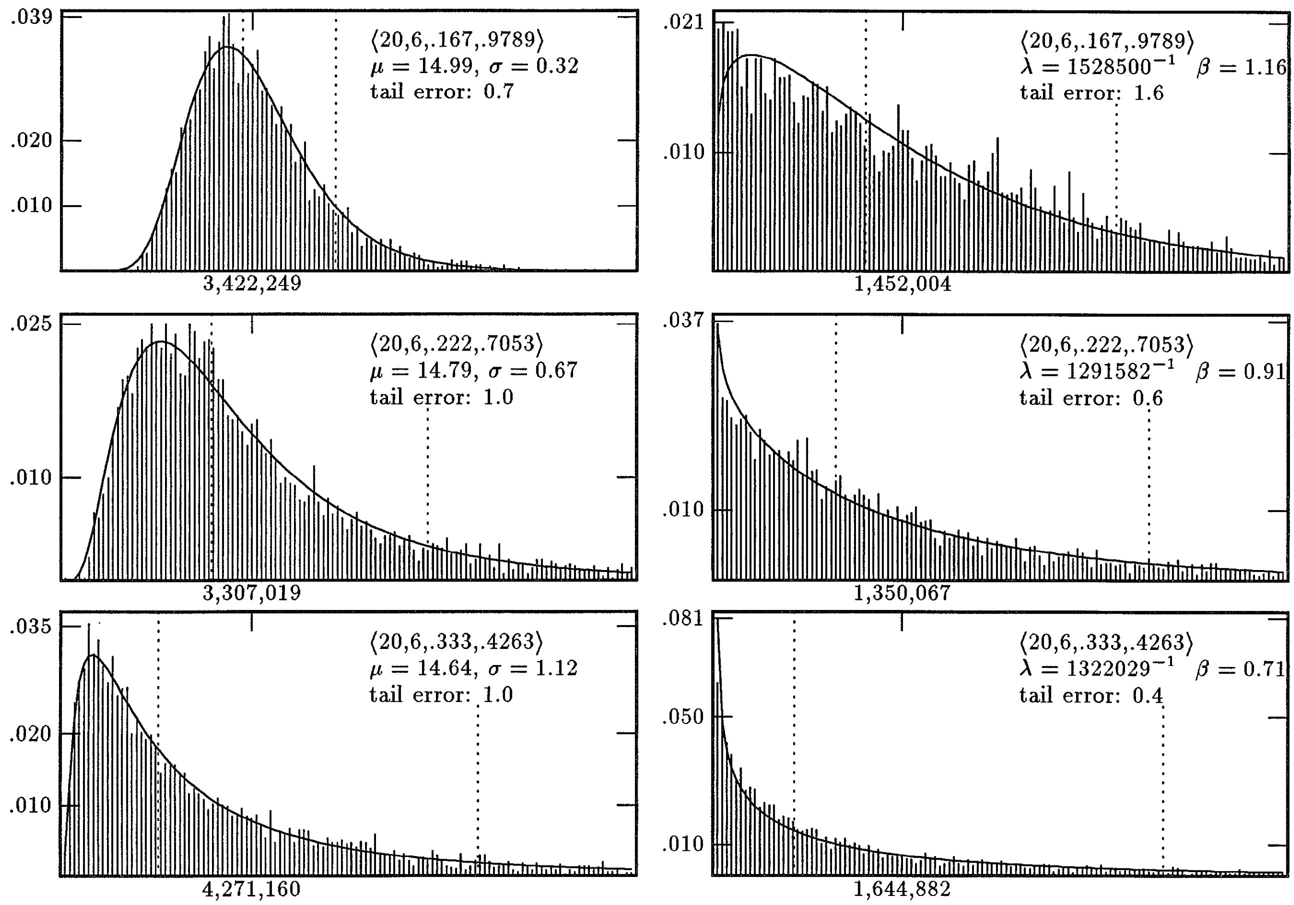

Figure 3: Experiments with a simple backtracking algorithm. See caption of Fig. 1 for notes on the graphs.

constraint tightness. We also report in Fig. 4 on problems with more variables, values of D other than 6, and the more complex BJ+DVO algorithm. Experiments with 3SAT problems, with and without variable ordering heuristics, indicate that the Weibull and lognormal distributions can model non-binary problems as well.

As the visual evidence in Fig. 1 and Fig. 3 indicates, the Weibull distribution does not provide a close fit for solvable problems. The fit from about the median rightwards is reasonably good, but the frequencies of the easiest problems are not captured. On CSPs with relatively dense constraint graphs (e.g. $\langle 50, 6, .167, .3722\rangle$ with BJ+DVO and $\langle 20, 6, .167, .9789\rangle$ with BT) $\beta > 1$ causes a peak in the curve which does not reflect the data. When the number of constraints is relatively small and the constraints themselves fairly tight (e.g. $\langle 50, 6, .500, .0833\rangle$ with BJ+DVO and $\langle 20, 6, .500, .4263\rangle$ with BT), the peak of the Weibull curve with $\beta < 1$ is much higher than what is observed experimentally.

In addition to consistency checks, we recorded CPU seconds and number of nodes in the search tree explored, and found that using those measures resulted in almost identical goodness-of-fit.

5. Discussion

The widespread use of the Weibull and lognormal distributions in reliability theory suggests that concepts from that field may be useful in understanding CSP search. An important notion in reliability is the failure or hazard rate, defined as $h(t) = f(t)/(1-F(t))$, where $f(t)$ and $F(t)$ are the density function and cdf. In CSP solving, we might call this rate the completion rate. If a problem is not solved at time t, $h(t)\cdot\Delta t$ is the probability of completing the search in $(t, t+\Delta t)$. For the exponential distribution, $h(t) = \lambda$ is constant. The completion rate of the Weibull distribution is $h(t) = \lambda^{\beta}\beta t^{\beta-1}$ which increases with t if $\beta > 1$ and decreases with t for $\beta < 1$. Thus when $\beta < 1$, each consistency check has a smaller probability of being the last one than the one before it. For the lognormal distribution no closed form expression of $h(t)$ exists. Its completion rate is nonmonotone, first increasing and then decreasing to 0. For $\sigma \approx 0.5$, $h(t)$ is very roughly constant, as the

⟨ N, D, T, C ⟩ Parameters	Unsolvable / Lognormal Mean	μ	σ	Tail
Algorithm: BT				
⟨20, 4, .125, .9895⟩	425	12.88	0.41	0.7
⟨20, 4, .250, .4421⟩	407	**12.54**	**0.88**	1.0
⟨20, 4, .375, .2579⟩	633	**12.58**	**1.25**	1.1
⟨20, 4, .500, .1579⟩	1,888	13.16	1.61	1.1
⟨20, 6, .167, .9789⟩	3,422	**14.99**	**0.32**	0.7
⟨20, 6, .222, .7053⟩	3,307	**14.79**	**0.67**	1.0
⟨20, 6, .333, .4263⟩	4,271	**14.64**	**1.12**	1.0
⟨20, 6, .500, .2316⟩	15,079	15.20	1.63	1.0
⟨20, 10, .210, 1.00⟩	54,024	17.79	0.17	0.5
⟨20, 10, .280, .7158⟩	57,890	17.53	0.83	0.7
⟨20, 10, .410, .4368⟩	94,242	17.43	1.36	0.8
Algorithm: BJ				
⟨20, 6, .167, .9789⟩	1,086	**13.86**	**0.26**	0.8
⟨20, 6, .222, .7053⟩	769	**13.40**	**0.54**	0.9
⟨20, 6, .333, .4263⟩	452	**12.61**	**0.90**	0.9
⟨20, 6, .500, .2316⟩	244	11.56	1.30	0.8
⟨25, 6, .167, .7667⟩	8,390	**15.82**	**0.49**	0.8
⟨25, 6, .222, .5533⟩	5,446	**15.25**	**0.73**	1.0
⟨25, 6, .333, .3333⟩	3,337	**14.42**	**1.10**	0.9
⟨25, 6, .500, .1800⟩	1,548	13.04	1.55	1.0
Algorithm: BT+MW				
⟨20, 6, .167, .9789⟩	2,755	**14.79**	**0.29**	0.8
⟨20, 6, .222, .7053⟩	358	12.71	0.40	0.8
⟨20, 6, .333, .4263⟩	53	**10.70**	**0.61**	0.9
⟨20, 6, .500, .2316⟩	9	**8.70**	**0.88**	0.8
⟨30, 6, .167, .6345⟩	23,735	**16.85**	**0.52**	0.9
⟨30, 6, .222, .4552⟩	4,909	**15.19**	**0.66**	1.1
⟨30, 6, .333, .2713⟩	592	**12.88**	**0.91**	1.0
⟨30, 6, .500, .1494⟩	65	**10.34**	**1.21**	1.0
Algorithm: FC				
⟨20, 6, .167, .9789⟩	251	12.41	0.22	0.6
⟨20, 6, .222, .7053⟩	173	11.96	0.46	0.8
⟨20, 6, .333, .4263⟩	110	**11.30**	**0.79**	1.0
⟨20, 6, .500, .2316⟩	122	**10.88**	**1.29**	1.1
Algorithm: BJ+DVO				
⟨50, 6, .167, .3722⟩	1,662	**14.27**	**0.33**	0.8
⟨50, 6, .222, .2653⟩	534	**13.09**	**0.43**	1.0
⟨50, 6, .333, .1576⟩	81	**11.10**	**0.63**	1.0
⟨50, 6, .500, .0833⟩	9	8.65	0.92	0.7
⟨75, 6, .333, .1038⟩	777	13.25	0.79	1.5
⟨75, 6, .500, .0544⟩	48	9.98	1.27	1.0
⟨30, 6, .333, .2713⟩	12	**9.28**	**0.42**	1.0
⟨40, 6, .333, .2000⟩	30	**10.18**	**0.54**	1.0
⟨60, 6, .333, .1305⟩	201	**11.96**	**0.71**	1.3
⟨150, 3, .222, .0421⟩	39	**10.05**	**1.02**	1.1
3SAT using DP (units are .01 CPU seconds)				
50 vars, 218 clauses	297	**5.51**	**0.61**	1.0
70 vars, 303 clauses	3,079	**7.78**	**0.71**	0.8
3SAT using DP+HEU (units are .01 CPU seconds)				
50 vars, 218 clauses	38	3.60	0.30	1.1
70 vars, 303 clauses	105	4.60	0.32	1.2
100 vars, 430 clauses	787	6.60	0.37	1.1
125 vars, 536 clauses	3,246	**8.02**	**0.35**	1.1

Figure 4: Unsolvable problems and the lognormal stribution. Parameters at .95 significance level in bold.

rate of increase and then decrease are small. When $\sigma > 1.0$, $h(t)$ increases rapidly for very small values of t, and then decreases slowly.

Viewing the CSP solving task as a process with a decreasing completion rate and therefore a long tail provides a new perspective on extremely hard instances encountered amidst mostly easy problems. The easy and hard problems are two sides of the same coin. A decreasing $h(t)$ implies that many problems are completed quickly, since the density function is relatively high when t is low. Problems that are not solved early are likely to take a long time, as the completion rate is low for high t. It will be interesting to see if future studies show a Weibull-like distribution for underconstrained problems, where the extremely hard instance phenomenon is more pronounced (Hogg & Williams 1994; Gent & Walsh 1994).

Knowledge of the completion rate function can be used in resource-limited situations to suggest an optimum time-bound for an algorithm to process a single instance. Examples would be running multiple algorithms on a single instance in a time-sliced manner, as proposed in (Huberman, Lukose, & Hogg 1997), and environments where the goal is to complete as many problems as possible in a fixed time period.

We also observe a pattern that holds for both solvable and unsolvable problems: the sparser the constraint graph, the greater the variance of the distribution, indicated by larger σ and smaller λ. The effect is visible in Fig. 4, when comparing rows with the same N, D, and algorithm. Parameters T and C are inversely related at the 50% satisfiable point, so the effect may be due to increasing T as well. But we note that when D=6 and T=.333, with BJ+DVO, and N is increased, C and σ both change. Experiments with $\langle 50, 6, .222, .2653\rangle$ and $\langle 30, 6, .333, .2713\rangle$ have nearly identical values for C and σ. This leads us to believe that variation in the graph density parameter, C, is primarily responsible for the variation in the shape of the distribution, for a given algorithm. The pattern holds even with BT. BJ, FC, and BT+MW can exploit tight constraints and a sparse graph to make such problems much easier. BT does not, but we still find greater variance with lower C.

In addition to the lognormal and Weibull distributions, we also investigated several other standard continuous probability distributions. We found the inverse Gaussian distribution to be almost identical to the lognormal in many cases, but in experiments on problems with relatively tight constraints and sparse graphs (e.g. $\langle 50, 6, .500, .0833\rangle$), the inverse Gaussian tended to be much too high at the mode. Also, its fit to the data on the right tail, as measured by our tail error statistic, was inferior. The gamma distribution is another candidate for modelling solvable problems. It usually fit the data a bit less well than the Weibull, and tended to show too high probability in the right tail.

6. Related work

Mitchell (1994) shows results from a set of experiments in which the run time mean, standard deviation, and maximum value all increase as more and more samples are recorded. This result is entirely consistent with the Weibull and lognormal distributions, as both tend to have long tails and high variance. Hogg and Williams (1994) provide an analytical analysis of the exponentially long tail of CSP hardness distributions. Their work suggests that the distributions at the 50% satisfiable point are quite different than the distributions elsewhere in the parameter space. Selman and Kirkpatrick (1996) have noted and analyzed the differing distributions of satisfiable and unsatisfiable instances. Kwan (1996) has recently shown empirical evidence that the hardness of randomly generated CSPs and 3-coloring problems is not distributed normally.

7. Conclusions

We have shown that for random CSPs generated at the 50% solvable point, the distribution of hardness can be summarized by two continuous probability distribution functions, the Weibull distribution for solvable problems and the lognormal distribution for unsolvable problems. The goodness-of-fit is generally statistically significant at the .95 level for the unsolvable problems, but only approximate for the solvable problems. The fit of distribution to data is equally good over a variety of backtracking based algorithms. Employing this approach will permit a more informative method of reporting experimental results. It may also lead to more statistically rigorous comparisons of algorithms, and to the ability to infer more about an algorithm's behavior from a smaller size test than was previously possible.

This study can be continued in several directions: to different problem generators, to parameters not at the 50% satisfiable point, and to a wider range of algorithms, particularly ones not derived from backtracking. We hope that further research into the distribution of CSP hardness will lead to both better reporting and better understanding of experiments in the field.

Acknowledgement

We thank Rina Dechter, Satish Iyengar, Eddie Schwalb, and the anonymous reviewers for many perceptive and helpful comments.

References

Aitchison, J., and Brown, J. A. C. 1957. *The Lognormal Distribution.* Cambridge, England: Cambridge University Press.

Bitner, J. R., and Reingold, E. 1975. Backtrack programming techniques. *Communications of the ACM* 18:651–656.

Crawford, J. M., and Auton, L. D. 1993. Experimental results on the crossover point in satisfiability problems. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 21–27.

D'Agostino, R. B., and Stephens, M. A. 1986. *Goodness-Of-Fit Techniques.* New York: Marcel Dekker, Inc.

Davis, M.; Logemann, G.; and Loveland, D. 1962. A Machine Program for Theorem Proving. *Communications of the ACM* 5:394–397.

Dechter, R. 1992. Constraint networks. In *Encyclopedia of Artificial Intelligence.* John Wiley & Sons, 2nd edition.

Freuder, E. C. 1982. A sufficient condition for backtrack-free search. *JACM* 21(11):958–965.

Frost, D., and Dechter, R. 1994. In search of the best constraint satisfaction search. In *Proceedings of the Twelfth National Conference on Artificial Intelligence.*

Gent, I. P., and Walsh, T. 1994. Easy problems are sometimes hard. *Artificial Intelligence* 70:335–345.

Haralick, R. M., and Elliott, G. L. 1980. Increasing Tree Search Efficiency for Constraint Satisfaction Problems. *Artificial Intelligence* 14:263–313.

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

Huberman, B. A.; Lukose, R. M.; and Hogg, T. 1997. An Economics Approach to Hard Computational Problems. *Science* 275:51–54.

Kwan, A. C. M. 1996. Validity of Normality Assumption in CSP Research. In *PRICAI'96: Topics in Artificial Intelligence. Proc. of the 4th Pacific Rim Int'l Conf. on Artificial Intelligence*, 253–263.

Mitchell, D.; Selman, B.; and Levesque, H. 1992. Hard and Easy Distributions of SAT Problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, 459–465.

Mitchell, D. 1994. Respecting Your Data (I). In *AAAI-94 Workshop on Experimental Evaluation of Reasoning and Search Methods*, 28–31.

Nelson, W. 1990. *Accelerated Testing: Statistical Models, Test Plans, and Data Analyses.* New York: John Wiley & Sons.

Prosser, P. 1993. Hybrid Algorithms for the Constraint Satisfaction Problem. *Computational Intelligence* 9(3):268–299.

Prosser, P. 1996. An empirical study of phase transitions in binary constraint satisfaction problems. *Artificial Intelligence* 81:81–109.

Selman, B., and Kirkpatrick, S. 1996. Critical behavior in the computational cost of satisfiability testing. *Artificial Intelligence* 81:273–295.

Smith, B. M., and Dyer, M. E. 1996. Locating the phase transition in binary constraint satisfaction problems. *Artificial Intelligence* 81:155–181.