# The Scaling of Search Cost*

**Ian P. Gent** and **Ewan MacIntyre** and **Patrick Prosser** and **Toby Walsh**
APES Research Group, Department of Computer Science,
University of Strathclyde, Glasgow G1 1XH, Scotland
Email: {ipg,em,pat,tw}@cs.strath.ac.uk

## Abstract

We show that a rescaled constrainedness parameter provides the basis for accurate numerical models of search cost for both backtracking and local search algorithms. In the past, the scaling of performance has been restricted to critically constrained problems at the phase transition. Here, we show how to extend models of search cost to the full width of the phase transition. This enables the direct comparison of algorithms on both under-constrained and over-constrained problems. We illustrate the generality of the approach using three different problem domains (satisfiability, constraint satisfaction and travelling salesperson problems) with both backtracking algorithms like the Davis-Putnam procedure and local search algorithms like GSAT. As well as modelling data from experiments, we give accurate *predictions* for results beyond the range of the experiments.

## Introduction

How does the performance of an algorithm scale? Theoretical analysis, especially average-case, can be very difficult. We often therefore use empirical data, especially random instances. For many NP-complete problems, the hardest random instances tend to cluster near a transition in solubility where problems are "critically constrained" (Cheeseman, Kanefsky, & Taylor 1991). A definition for the 'constrainedness' of problem ensembles has been proposed that predicts the location of this phase transition for many different NP-complete problem classes (Gent *et al.* 1996). A simple rescaling of this parameter models the scaling of the transition in solubility with problem size (Kirkpatrick & Selman 1994; Gent *et al.* 1996).

In this paper, we investigate claims made in (Gent *et al.* 1995; Selman & Kirkpatrick 1996) that this rescaled parameter may also be suitable for modelling search cost. We show that this parameter permits simple but accurate models of the growth of search cost across the full width of the phase transition – in the past, models have usually been restricted to growth at a single point, critically constrained problems with the maximum search cost. In these models, we take problem size to be the log of the size of the state space. As well as modelling empirical data, we predict search cost at problem sizes beyond the range of our experiments. We model both complete and local search procedures, and thereby compare algorithms that share no natural measure of search cost other than CPU time.

## Constrainedness

(Gent *et al.* 1996) proposes a general purpose parameter, $\kappa$, for the constrainedness of an ensemble of problems. If each problem has a state space with $2^N$ states, of which $\langle Sol \rangle$ are expected to be solutions, then

$$\kappa \quad =_{\text{def}} \quad 1 - \frac{\log_2(\langle Sol \rangle)}{N} \tag{1}$$

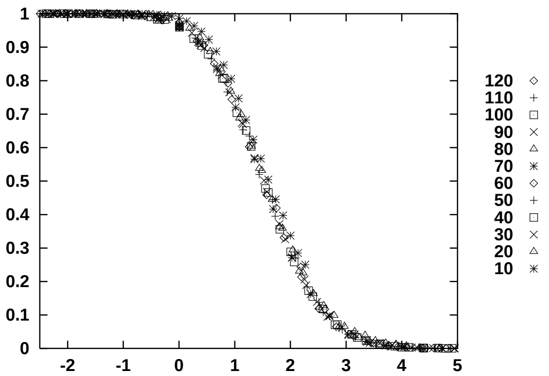
This definition gives parameters used to study phase transition behaviour in many problem classes including satisfiability, constraint satisfaction, number partitioning, and travelling salesperson problems. A phase transition in solubility is predicted to occur around $\kappa \approx 1$.

Whilst $\langle Sol \rangle$ can grow exponentially with problem size at the phase transition in solubility (for example, as $2^{0.18N}$ for random 3-SAT), $\kappa$ varies much more slowly. Indeed, a technique borrowed from statistical mechanics called finite size scaling (Barber 1983) can usually model any variation by a simple rescaling of $\kappa$. Around a critical value (in this case, $\kappa_c$) finite size scaling predicts that maroscopic properties of the ensemble (in this case, the probability of solubility) will be indistinguishable except for a simple power law scaling with problem size (in this case, $N^{1/\nu}$). Following (Gent *et al.* 1996), we define a rescaled parameter,

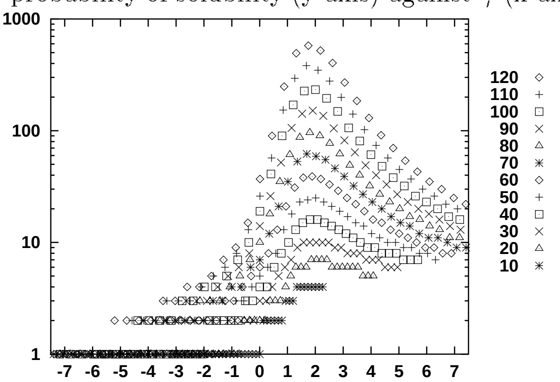
$$\gamma \quad =_{\text{def}} \quad \frac{\kappa - \kappa_c}{\kappa_c} N^{1/\nu} \tag{2}$$

* Supported by EPSRC awards GR/K/65706 and GR/L/24014, and a Strathclyde University starter grant. We thank Bob Craig and Mark Stickel for code, and Vince Darley for correcting an error in the value of $\kappa$ for the ATSP.

To illustrate this rescaling, we use a popular ensemble of CSP's described by the tuple $\langle n, m, p_1, p_2 \rangle$, as used in for example (Prosser 1996; Smith & Dyer 1996). Each problem has $n$ variables, a domain size of $m$, and $p_1 n(n-1)/2$ binary constraints, each with $p_2 m^2$ of the possible pairs of values ruled out. In Figure 1(a), we plot the probability of solubility for the ensemble $\langle n, 3, p_1, \frac{2}{9} \rangle$ against the rescaled parameter $\gamma$ with $N = n \log_2(3)$. We use 10,000 problems at each point. Analysis of the experimental data suggests values of 0.64 for $\kappa_c$ and 2.1 for $\nu$. We see a familiar transition in solubility around $\gamma \approx 0$. In Figure 1(b), we plot the median search cost for the algorithm FC-CBJ-FF, forward checking with conflict directed backjumping using the fail first heuristic (Prosser 1993), against $\gamma$ using the same values of $\kappa_c$ and $\nu$. We use leaf-nodes as a measure of search cost. This graph shows the classic "easy-hard-easy" pattern through the phase transition.




(a) probability of solubility (y-axis) against $\gamma$ (x-axis)




(b) median leaf nodes searched (y-axis) against $\gamma$

Figure 1: Rescaled phase transition for $\langle n, 3, p_1, \frac{2}{9} \rangle$ with $\kappa_c = 0.64$, $\nu = 2.1$

## Scaling of Search Cost

To model growth of search cost with problem size, we need to compute a well-defined measure at each problem size. Typically, this has been the maximum search cost, experienced on critically constrained problems at the phase transition. The rescaled constrainedness parameter allows us to build more detailed models in which we can in addition study growth of search cost on under and over-constrained problems away from the phase transition.

(Selman & Kirkpatrick 1996) suggests normalizing the maximum value of measures like median search cost. This is in effect a one parameter model since it predicts search cost at a given problem size when provided with the normalising constant. Figure 2 shows that this model is only accurate close to the peak in search cost. The equal separation between contours

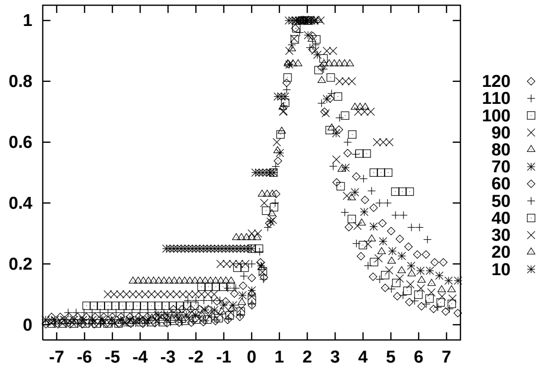
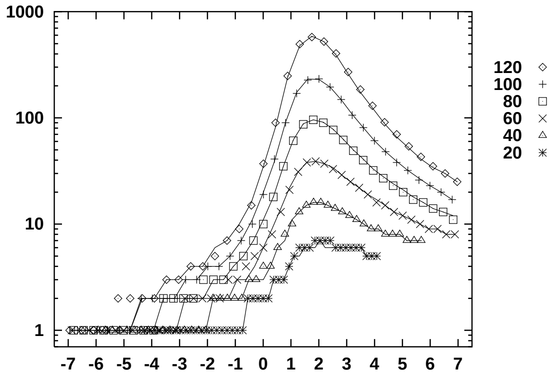


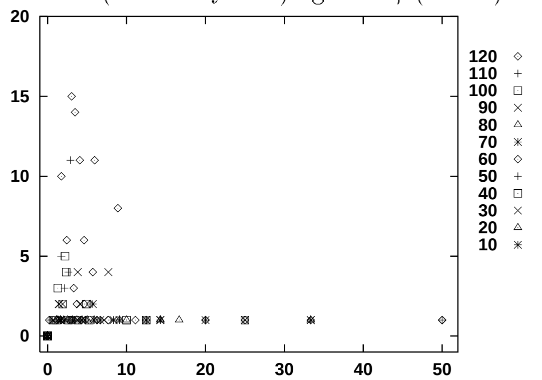

Figure 2: Normalised median leaf nodes searched (y-axis) for $\langle n, 3, p_1, \frac{2}{9} \rangle$ against $\gamma$ with $\kappa_c = 0.64$, $\nu = 2.1$

in Figure 1(b) suggests a simple two parameter model based on exponential growth. For a given percentile of search cost and value of $\gamma$, we model search cost as $\alpha 2^{\beta N}$, where $\alpha$ and $\beta$ are independent of $N$. To compute $\alpha$ and $\beta$, we use the least squares estimates for the intersection and gradient of the log of search cost against $N$, interpolating data points where necessary. This interpolation may introduce errors near the phase transition where search cost changes rapidly and the estimated values for the exponents, $\beta$, are largest.

Figure 3 shows that this model describes median search cost very successfully across the full width of the phase transition. In Figure 3(a), points give actual data whilst lines indicate the values that our model gives rounded to the nearest integer. We interpolate between adjacent estimates of the regression parameters where necessary. Figure 3(b) shows that our model gives *either* a very small error in the number of nodes *or* a small percentage error. The model is never more than 15 nodes out, and only as much as 10% in error when it also correct to within a single node. Indeed, ignoring the large number of points where the model is




(a) modelled (lines) and observed (points) search costs (both on y-axis) against $\gamma$ (x-axis)

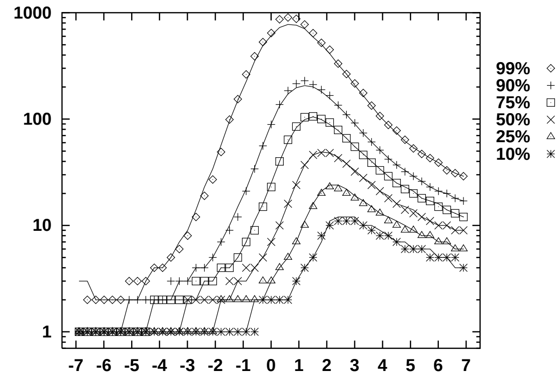



(b) absolute error (y-axis) against absolute percentage error (x-axis)

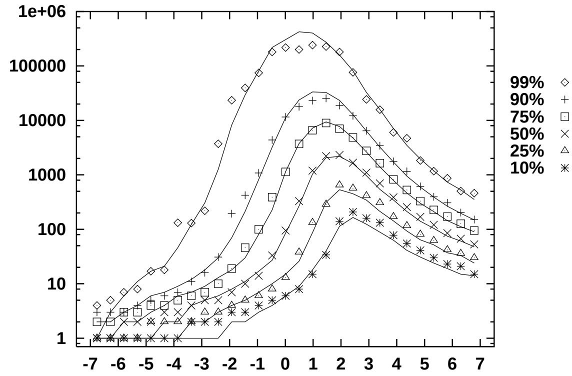
Figure 3: Median leaf nodes searched by FC-CBJ-FF on $\langle n, 3, p_1, \frac{2}{9}\rangle$.

*exact*, the overwhelming majority of the data is modelled to within both 10% and 10 nodes error. We obtain similar results with other percentiles of search cost.

Since we tested 12 values of $n$, we have modelled a system with 12 degrees of freedom to a reasonable degree of accuracy using just 2 parameters. Unlike Selman and Kirkpatrick's normalizing model, we can also predict search costs at new problem sizes. We first test predictions for a problem size within the range of our experimental data, namely 65 variables, again using 10,000 problems at each point. This value of $n$ was not previously tested and thus played no role in the regression. Figure 4(a) shows the accuracy of the predictions for a variety of percentiles at 65 variables. All data points shown were predicted to within either 15% or 10 nodes. We next test how our model extrapolates to problem sizes *larger* than used in the regression. In Figure 4(b) we plot the predicted and observed results for 150 variables using 1000 problems at each point. Although we now see significant errors of more than 100%, such large errors only occur in the 99th percentile near the phase transition. Because of the reduced sample size, the 99th percentile is subject to more noise than usual. The model both over- and under-estimates the observed 99th percentile. The lack of a consistent trend suggests experimental noise rather than a modelling error. In other percentiles, most predictions are accurate to within 25%. Note that, as these predictions are for problems with 30 more variables than used in the regression, the underlying search space has increased by a *factor* of more than $10^{14}$.




(a) $n = 65$




(b) $n = 150$

Figure 4: Predicted (lines) and observed (points) percentiles of leaf nodes searched (both on y-axis) for $\langle n, 3, p_1, \frac{2}{9}\rangle$ against $\gamma$ (x-axis).

Ensembles of problems close to the phase transition contain both soluble and insoluble problems. There are situations in which only one kind of problem is of interest. For example, local search procedures usually cannot show insolubility. Our methods work even better if we construct a model just for soluble or insoluble problems. In Figure 5 we use the same model as before but restricted to soluble or insoluble problems, depending on which was predominant. The worst errors of prediction are smaller with a separated model, as might be expected, since soluble and insoluble problems behave very differently. The worst errors in the unified model are in the 'mushy region' (Smith & Dyer 1996) where both soluble and insoluble problems occur. The model also works with soluble problems when most problems are insoluble, and insoluble problems when most are soluble. However, we must generate a large number of problems to ensure a sufficient sample of the rare soluble or insoluble problems.

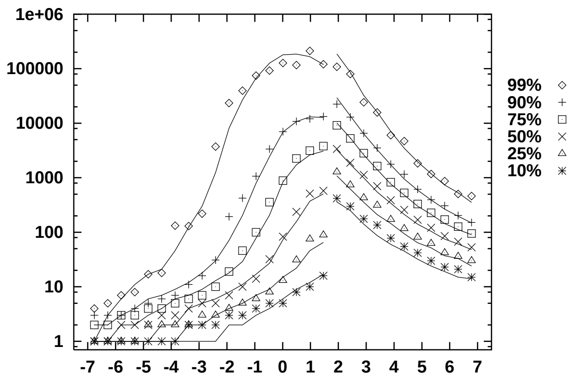



Figure 5: Predicted (lines) and observed (points) percentiles of leaf nodes searched (both on y-axis) for $\langle 150, 3, p_1, \frac{2}{9} \rangle$ against $\gamma$ (on x-axis). Where 50% or more of problems are (in)soluble, we study only the search cost of the (in)soluble problems.

We are also able to model a completely different ensemble of CSP's with a fixed number of variables and varying domain size. As in (Gent *et al.* 1995), we used $\langle 10, m, 1.0, p_2 \rangle$ with $m = 10$ to 100. Finite size scaling now requires $\kappa_c = 1.01$, $\nu = 0.8$ and $N = 10 \log_2(m)$. The same model of exponential growth accurately models observed median search costs and predicts values for problem sizes not previously tested.

## Travelling Salesperson Problem

We now consider a different NP-complere problem, the asymmetric travelling salesperson problem (ATSP) with inter-city distances drawn randomly from a normal distribution with mean $\mu$ and standard deviation $\sigma$. We focus on the decision problem of deciding if there is a tour of length $d$ or less which visits all $n$ cities. For convenience, we normalise the tour length to $\hat{d} = (d - n\mu)/\sigma\sqrt{n}$. Correcting an error in (Gent *et al.* 1996),

$$\kappa \quad = \quad \frac{\hat{d}^2 \log_2(e)/2 + \log_2(|\hat{d}|\sqrt{2\pi})}{\log_2(n-1)!}$$

For $n$=6 to 48 in steps of 6, we use 1000 pseudo-random problems with inter-city distances independently and normally distributed with $\mu$=$10^6$ and $\sigma$=$10^5$. We solve these problems using a branch and bound algorithm with the Hungarian heuristic (Carpaneto & Toth 1980) for branching. (Gent *et al.* 1996) shows that the probability of a tour existing for this problem class rescales with $\kappa_c = 0.75$, $\nu = 2$ and $N = \log_2(n-1)!$. As with constraint satisfaction problems, the measure most suitable for modelling growth of search cost appears to be leaf nodes. We only consider the high percentiles of leaf nodes, as many problems are solved in just one node. The same model of simple exponential growth fits the data well. We put aside the extremes of our data, and construct a model for $\gamma = -3$ to 3 in steps of 0.25 using $n$=12 to 42. We then compare the modelled data with observed, and also the search costs the model predicts at 6 and 48 cities. Figure 6 shows the accuracy of the modelling and predictions. For example, the model successfully predicts that at $n = 6$ we will explore just a single leaf node throughout the phase transition. It is remarkable that a simple model of exponential growth does so well when search costs never reaches very large values, and granularity therefore plays an important role. (Gent & Walsh 1996) reports a phase transition in the Euclidean TSP, but the parameter used was derived from asymptotic considerations and not from the constrainedness parameter $\kappa$. It would be interesting to see if modelling of search cost works well against this parameter.

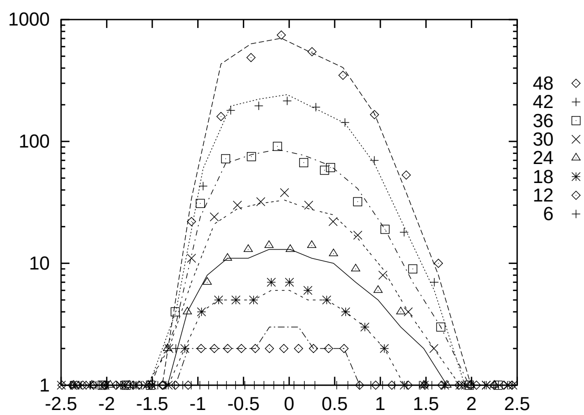



Figure 6: Modelled/predicted (lines) and observed (points) 90th percentile of leaf nodes searched to solve ATSP instances (both on y-axis) against $\gamma$ (x-axis) with $\kappa_c = 0.75$, $\nu = 2$.

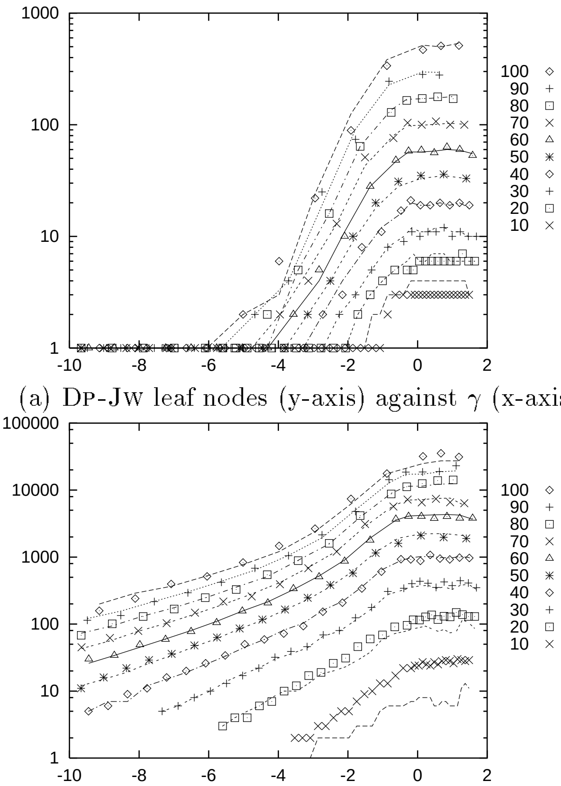
## Local vs Complete Search

By modelling the growth of search costs for complete and local search procedures, we can compare algorithms that only share CPU time as a measure of search cost. We use satisfiability (Sat) as a case study for such a comparison. As in many previous studies, for example (Selman, Levesque, & Mitchell 1992), we use the random 3-Sat problem class. For problems from this class with $l$ clauses and $n$ variables, $\kappa = -\log_2(7/8)\ \frac{l}{n}$ (Gent *et al.* 1996). This is just a constant times the more familiar ratio $l/n$. Once again finite size scaling (of the probability of satisfiability) works well with the critical value of $l/n$ being 4.17, corresponding to $\kappa_c = 0.803$, $\nu = 1.5$ and $N = n$ (Kirkpatrick & Selman 1994).

For the complete procedure, we use the Davis-Putnam algorithm without pure literal deletion (Davis, Logemann, & Loveland 1962) with the branching heuristic from (Jeroslow & Wang 1990). We call this Dp-Jw. As in the previous sections, we measure leaf nodes searched. For the local search procedure, we use the random hill-climbing algorithm Gsat (Selman, Levesque, & Mitchell 1992) measuring, as in previous studies, the number of truth-values flipped during search. As Gsat cannot solve unsatisfiable problems, we restrict analysis for both algorithms to satisfiable problems. From $n = 10$ to 100 and $l/n$ up to at least 4.4, we tested Gsat and Dp-Jw on 1000 satisfiable pseudo-random 3-Sat problems. For Gsat, we use the value of 'max-flips' that minimises the number of flips used at $l/n = 4.3$ for each $n$.

Figure 7(a) shows that the simple exponential growth model also works well for Dp-Jw. Since we set aside $n = 100$, the top contour is a prediction rather than modelling. We were unable to fit the data for Gsat to the exponential model of search growth. Instead, we construct a simple two parameter power law model in which the number of flips used grows as $an^b$ at each value of $\gamma$. Figure 7(b) shows that this model describes the data well when we set aside $n = 10$, 20, and 100. The scaling exponent $b$ is never greater than 4, i.e. search never grows faster than quartically. The lines at $n = 10$, 20, and 100 are predictions. The predictions are not very accurate for $n = 10$ or 20, but are good for $n = 100$ though they seem to under-estimate the true values near the phase transition. (Parkes & Walser 1996) report being unable to fit data (for the peak in search cost only) for a similar algorithm, Wsat, using a two parameter model.

Note that the same technique worked well for both complete and local search procedures: i.e. modelling the growth of search cost against the rescaled constrainedness parameter $\gamma$ and the problem size. The differences between the models suggest a different scaling of search cost for the two algorithms. While search cost for Dp-Jw seems to grow exponentially, search cost in Gsat appears to grow no faster than $n^4$. This suggests that Gsat scales better than Dp-Jw. This was not motivated by comparison of pure run-times, as in (Selman, Levesque, & Mitchell 1992). At 100 variables our code for each algorithm solved problems in broadly similar amounts of cpu time. Nor was it motivated by naive comparison of the different search measures: Gsat needs many more flips than Dp-Jw needs leaf nodes at 100 variables. The suggestion arises out of a simple mathematical model, constructed from data about comparatively small problems. Whilst it is valuable to test problems as large as technically feasible, our use of small problems enabled large samples to be used across the phase transition.




(a) Dp-Jw leaf nodes (y-axis) against $\gamma$ (x-axis)



(b) Gsat flips (y-axis) against $\gamma$ (x-axis)

Figure 7: Modelled/predicted (lines) and observed (points) 90th percentile of the search cost for random 3-Sat with $\kappa_c = 0.803$ and $\nu = 1.5$

## Related work

Gent and Walsh report a possible cubic growth in search cost for Gsat at the random 3-Sat phase transition (Gent & Walsh 1993). Parkes and Walser also record sub-exponential growth for several variants of the Gsat procedure at the random 3-Sat phase transition (Parkes & Walser 1996). All these results were just for the worst case, critically constrained problems at the phase transition. By comparsion, this paper models scaling across the entire phase transition.

Crawford and Auton report simple exponential growth for TABLEAU, a Davis Putnam variant on the random 3-SAT problem class at $l/n = 4.2$ (the phase transition) and $l/n = 10$ (over-constrained region) (Crawford & Auton 1996). They also record roughly linear growth at $l/n = 1, 2$ and 3 in the under-constrained region, but caution that some problems in this region can be as hard as those at the phase transition. Selman and Kirkpatrick construct a simple one parameter model that normalizes maximum search costs for a Davis Putnam procedure on the random 3-SAT problem class, using the ratio of clauses to variables rescaled by finite size scaling (Selman & Kirkpatrick 1996). Whilst this model performs well close to the phase boundary, it performs poorly on under- and over-constrained problems.

Theoretical analysis of performance of algorithms for satisfiability has largely been restricted to the easier constant probability problem class. One exception is Yugami who has developed an approximate theoretical model for the average case complexity of the Davis-Putnam procedure across the random 3-SAT phase transition (Yugami 1995). Whilst the results agree well with experiment, the derivation is complex and only yields recursion equations.

## Conclusions

We have proposed a very general technique for studying the growth of search cost in random NP-complete problems. For a given problem class, the scaling of the phase transition in the probability of solubility suggests a rescaled constrainedness parameter. This parameter then provides the basis for models of the growth of search cost with problem size. We have shown that this technique works in a variety of different NP-complete problem classes such as constraint satisfaction, travelling salesperson, and satisfiability problems. For the first time, this enables empirically derived predictions of search cost across the whole phase transition. We have shown this modelling technique works for both complete and local search procedures. We can thereby compare algorithms which have no comparable measure of search cost except for CPU time.

## References


Barber, M. N. 1983. Finite-size scaling. In *Phase Transitions and Critical Phenomena, Volume 8*, 145–266. Academic Press.

Carpaneto, G., and Toth, P. 1980. New branching and bounding criteria for the asymmetric travelling salesman problem. *Management Sci.* 26:736–743.

Cheeseman, P.; Kanefsky, B.; and Taylor, W. 1991. Where the really hard problems are. In *Proceedings of the 12th IJCAI*, 331–337.

Crawford, J., and Auton, L. 1996. Experimental results on the crossover point in random 3-SAT. *Artificial Intelligence* 81:31–57.

Davis, M.; Logemann, G.; and Loveland, D. 1962. A machine program for theorem-proving. *Comms. ACM* 5:394–397.

Gent, I. P., and Walsh, T. 1993. Towards an understanding of hill-climbing procedures for SAT. In *Proceedings of AAAI-93*, 28–33.

Gent, I., and Walsh, T. 1996. The TSP phase transition. *Artificial Intelligence* 88:349–358.

Gent, I.; MacIntyre, E.; Prosser, P.; and Walsh, T. 1995. Scaling effects in the CSP phase transition. In *Principles and Practice of Constraint Programming*, 70–87. Springer.

Gent, I.; MacIntyre, E.; Prosser, P.; and Walsh, T. 1996. The constrainedness of search. In *Proceedings of AAAI-96*, 246–252.

Jeroslow, R. E., and Wang, J. 1990. Solving propositional satisfiability problems. *Annals of Mathematics and Artificial Intelligence* 1:167–187.

Kirkpatrick, S., and Selman, B. 1994. Critical behavior in the satisfiability of random boolean expressions. *Science* 264:1297–1301.

Parkes, A., and Walser, J. 1996. Tuning local search for satisfiability testing. In *Proceedings of AAAI-96*, 356–362.

Prosser, P. 1993. Hybrid algorithms for the constraint satisfaction problem. *Computational Intelligence* 9:268–299.

Prosser, P. 1996. An empirical study of phase transitions in binary constraint satisfaction problems. *Artificial Intelligence* 81:127–154.

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

Selman, B.; Levesque, H.; and Mitchell, D. 1992. A new method for solving hard satisfiability problems. In *Proceedings of AAAI-92*, 459–465.

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

Yugami, N. 1995. Theoretical analysis of Davis-Putnam procedure and propositional satisfiability. In *Proceedings of IJCAI-95*, 282–288.