The Constrainedness of Search[*]

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

Toby Walsh
IRST, I38100 Trento &
DIST, I16145 Genova, Italy
Email: toby@itc.it

Abstract

We introduce a parameter that measures the "constrainedness" of an ensemble of combinatorial problems. If problems are over-constrained, they are likely to be insoluble. If problems are under-constrained, they are likely to be soluble. This constrainedness parameter generalizes a number of parameters previously used in different NP-complete problem classes. Phase transitions in different NP classes can thus be directly compared. This parameter can also be used in a heuristic to guide search. The heuristic captures the intuition of making the most constrained choice first, since it is often useful to branch into the least constrained subproblem. Many widely disparate heuristics can be seen as minimizing constrainedness.

Introduction

Will a problem be soluble or insoluble? Will it be hard or easy? How can we develop heuristics for new problem domains? All these questions have been the subject of intensive study in recent years in a large number of problem domains including for example satisfiability, graph colouring, constraint satisfaction problems, and hamiltonian circuits (Cheeseman, Kanefsky, & Taylor 1991; Mitchell, Selman, & Levesque 1992; Williams & Hogg 1994; Smith & Dyer 1996). Here, we introduce some general methods which help to answer these questions in a wide range of problems. These methods are based on a definition of the constrainedness of an ensemble of combinatorial problems.

Problems which are very over-constrained are insoluble and it is usually easy to determine this. Problems which are very under-constrained are soluble and it is usually easy to guess one of the many solutions. A phase transition tends to occur inbetween when problems are "*critically constrained*" and it is difficult to determine if they are soluble or not (Cheeseman, Kanefsky, & Taylor 1991). As problem size increases, this phase transition occurs more rapidly as we vary the constrainedness. Surprisingly, we can often characterize an ensemble of problems using just two parameters: their size, and their constrainedness.

We will first define a parameter κ (the Greek letter 'kappa') which we use to measure constrainedness. We show how it can be calculated in several classes of combinatorial problems where phase transitions have been intensively studied, and also one where they have not. Then we show that κ can be used in a heuristic to guide search, and that it can outperform a good heuristic for constraint satisfaction problems. Finally we show how many existing heuristics for combinatorial search can be understood as heuristics to minimize κ.

Constrainedness

By studying the parameters introduced in a variety of domains like graph colouring and satisfiability, we propose a definition of the "constrainedness" of an ensemble of problems. We assume that each problem in an ensemble has a state space S with $|S|$ elements and a number, Sol of these states are solutions. Any point in the state space can be represented by a N-bit binary vector where $N = \log_2(|S|)$. Let $\langle Sol \rangle$ be the expected number of solutions averaged over the ensemble. We will define constrainedness, κ, of an ensemble by,

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

It is very important to note that this defines the constrainedness of an *ensemble* of problems, not of an individual problem. In following sections, we will show that this definition generalizes, unifies, and extends a large body of previous work on randomly generated problems. In such cases, the method of generation defines the ensemble of problems, so κ is well defined.

Further motivation for this definition comes from considering the probability that we can set the bits in the N-bit binary vector and arrive at a solution. If the problems are under-constrained, there will be many solutions and the probability that a bit can be set correctly will be high. If problems are over-constrained, there are very few or no solutions, and the probability that a bit can be set correctly will be much lower.

[*] Supported by an HCM personal fellowship to the last author, a Strathclyde University starter grant to the first author, and an EPSRC ROPA award GR/K/65706. We thank Bob Craig for his TSP solving code. For very helpful comments, we thank AAAI reviewers, Jeremy Frank, and the members of APES, especially Barbara Smith.

Constrainedness is thus closely related to this probability. Unfortunately, it is difficult to compute this probability directly. We do know, however, the probability that all the bits can be set correctly as this is just the expected number of solutions divided by the size of the state space. This equals the product of the N probabilities for each bit. In the absence of any other evidence, we assume that all these probabilities are equal. That is, we estimate a single probability by the geometric mean. We then take κ as the negative logarithm of this estimate.

Since $\log_2(\langle Sol \rangle) \leq N$, κ is bounded in the range $[0, \infty)$. A value $\kappa = 0$ corresponds to a completely unconstrained ensemble with $Sol = |S|$, while a value of $\kappa = \infty$ corresponds to a completely constrained ensemble with $Sol = 0$. In constraint satisfaction, (Williams & Hogg 1994; Smith & Dyer 1996) predict a phase transition between soluble and insoluble problems when $\langle Sol \rangle \approx 1$. In line with this prediction, using (1) we conjecture that a transition will occur when $\kappa \approx 1$. If $\kappa < 1$, problems are under-constrained and are typically soluble. If $\kappa > 1$, problems are over-constrained and are typically insoluble. The equality $\kappa \approx 1$ only gives a first approximation of the location of the phase transition: we will see that κ is typically between about 0.75 and 1 at the phase transition. More refined estimates take account of the variance in the number of solutions at the phase boundary (Williams & Hogg 1994; Smith & Dyer 1996).

There is no difference between the prediction of a phase transition at $\kappa \approx 1$ or at $\langle Sol \rangle \approx 1$. The value of the definition of κ is as a parameter for measuring problems. While $\langle Sol \rangle$ can grow exponentially with N, we will see that the value of κ at the phase transition varies very little. What variation there is can be modelled by the technique of finite size scaling. Indeed, κ is such a good parameter that it has been used independently in a number of problem classes, as we show below. For the first time, we can see that these assorted parameters all measure the same thing.

An example

It is often quite straightforward to compute $\langle Sol \rangle$ and therefore κ. Consider, for example, constraint satisfaction problems (CSP's). Each variable $v \in V$, has a domain of values M_v of size m_v. Each constraint $c \in C$ of arity a restricts a tuple of variables $\langle v_1, \ldots, v_a \rangle$, and rules out some proportion p_c of possible values from the cartesian product $M_{v_1} \times \ldots \times M_{v_a}$. We call p_c the "tightness" of a constraint. To avoid trivial problems we insist that all arities are at least one, but make no further restrictions. Problems may have variables with many different domain sizes, and constraints of many different arities and tightnesses.

The state space has size $\prod_{v \in V} m_v$. Each constraint rules out a proportion p_c of these states, so we have

$$\langle Sol \rangle = (\prod_{v \in V} m_v) \times (\prod_{c \in C} (1 - p_c))$$

Substituting this into (1) gives

$$\kappa \quad = \quad \frac{-\sum_{c \in C} \log_2(1 - p_c)}{\sum_{v \in V} \log_2(m_v)} \qquad (2)$$

(Williams & Hogg 1994) present a similar model for constraint satisfaction problems. The approach presented here can, as we show later, be applied to other problems of a very different nature like number partitioning and the travelling salesman problem.

Comparison with existing parameters

The definition of constrainedness generalizes several parameters introduced recently for satisfiability, constraint satisfaction, graph colouring, and number partitioning. We predict that it will prove useful in many other domains.

In satisfiability, we are given a formula with n variables and l clauses each of which has a literals. We can view this as an instance of a constraint satisfaction problem. Each variable has two values, true and false, so $m_v = 2$ for all v. Each clause rules out one of the 2^a possible tuples of values of the variables in the clause. So each clause is a constraint of tightness $1/2^a$, and there are l such constraints. Equation (2) gives

$$\kappa = -\log_2(1 - \frac{1}{2^a}) \frac{l}{n}$$

That is, a constant times l/n for fixed a. The ratio l/n has been used as an "order parameter" for satisfiability. A phase transition in satisfiability occurs around a critical value of l/n (Mitchell, Selman, & Levesque 1992). For large a, this phase transition occurs at a value close to $-1/\log_2(1 - \frac{1}{2^a})$ (Kirkpatrick & Selman 1994), that is around $\kappa \approx 1$, as expected.

In graph colouring, we are given a graph with n nodes and e edges, and wish to colour it with m colours. As a CSP, each node represents a variable with a fixed domain of size m, and each edge represents a binary constraint ruling out m of the m^2 possible pairs of colours, a tightness of $1/m$. So (2) gives

$$\kappa = \frac{e}{n} \log_m(\frac{m}{m-1})$$

This is a constant, namely $\log_m(\frac{m}{m-1})/2$, times the average degree of a node in the graph. The average degree has been used as an order parameter for describing the phase transition in colouring problems (Cheeseman, Kanefsky, & Taylor 1991). A phase transition has been observed in random 3-colouring problems at an average degree of 4.6 (Hogg & Williams 1994), corresponding to $\kappa = 0.84$.

In binary CSP's (in which constraints only have a binary arity), a standard means of generating test problems is to have n variables each with the same domain size of m. Given a constraint density of p_1, exactly $p_1 n(n-1)/2$ constraints are chosen, each with a tightness of p_2 (Prosser 1996; Smith & Dyer 1996). Such problems are described by the tuple, $\langle n, m, p_1, p_2 \rangle$. Using these values, (2) gives

$$\kappa = \frac{n-1}{2} p_1 \log_m(\frac{1}{1-p_2})$$

This has been used as a parameter for binary constraint satisfaction problems (Gent *et al.* 1995). The phase transition again occurs around $\kappa \approx 1$.

In number partitioning, we have n numbers from the range $(0, l]$ and wish to find an exact partition into m bags with the same sum. We have $N = n \log_2 m$ as there are m^n possible partitions. (Gent & Walsh 1996) present an "annealed" theory in which the expected number of exact partitions is

$$\langle Sol \rangle \approx m^n (\frac{1}{2})^{(m-1)\log_2(l)}$$

Although (2) no longer applies, substituting N and $\langle Sol \rangle$ into (1) gives

$$\kappa = (m-1)\frac{\log_m(l)}{n} \qquad (3)$$

In two-way partitioning, i.e. $m = 2$, a phase transition in solubility occurs at $\kappa = 0.96$ (Gent & Walsh 1996).

We thus see that our definition of κ generalises a number of parameters introduced in a variety of problem classes. This suggests that "constrainedness" is a fundamental property of problem ensembles. In addition to unifying existing parameters, we can now compare problems between classes. For example, the phase transition in 3-satisfiability problems occurs at $l/n = 4.24$ (Crawford & Auton 1993) which corresponds to $\kappa = 0.82$, roughly comparable to that in 3-colouring at $\kappa = 0.84$, while the phase transition in number partitioning occurs at $\kappa = 0.96$. This suggests that number partitioning problems at the phase transition may in some sense be more constrained. The definition of κ also allows us to treat a wider range of problems within a class. For example, we now deal with problems having mixed arity constraints, mixed domain sizes and mixed constraint tightnesses. This permits the computation of κ during search as domain sizes change and constraints are removed. We will see the value of this in a later section.

The travelling salesman problem

We now give a case study of using our definition of constrainedness in a new problem class. We consider the asymmetric travelling salesman problem (ATSP) with inter-city distances drawn from a normal distribution with mean μ and standard deviation σ. We consider the decision problem of determining if there is a tour of length d or less which visits all n cities. Most computational studies of the travelling salesman problem have been on the optimisation rather than the decision problem (Cheeseman, Kanefsky, & Taylor 1991; Zhang & Korf 1992). Although a phase transition has been observed in the decision problem in the two-dimensional Euclidean TSP (Gent & Walsh 1995b), the parameter used was based on an asymptotic result and its relationship to constrainedness is, as yet, uncertain.

The state space S contains all $(n-1)!$ possible distinct tours (one city is designated the starting point arbitrarily). Each of these tours has some length l. As the sum of n normal distributions, l has a normal distribution with mean $n\mu$ and standard deviation $\sigma\sqrt{n}$. If we normalise l to $\hat{l} = (l - n\mu)/\sigma\sqrt{n}$ then $\hat{l}$ is distributed normally with mean 0 and standard deviation 1. The probability that a randomly chosen tour has a length l less than or equal to some given length d is

$$prob(l \leq d) = \int_{-\infty}^{\hat{d}} \frac{e^{-x^2/2}}{\sqrt{2\pi}} dx$$

For $z < 0$ (Abramowitz & Stegun) gives the equality

$$\int_{-\infty}^{z} \frac{e^{-x^2/2}}{\sqrt{2\pi}} dx = \frac{e^{-z^2/2}}{\sqrt{2\pi}} \left(\frac{1}{|z|} + O(\frac{1}{|z|^3}) \right)$$

The optimal tour length will tend to have $\hat{d} << 0$ so the error term will be small. Accordingly we use the approximation

$$prob(l \leq d) \approx \frac{e^{-\hat{d}^2/2}}{|\hat{d}|\sqrt{2\pi}}$$

Multiplying this by $(n-1)!$, the number of distinct tours, gives $\langle Sol \rangle$, the expected number of tours less than or equal to d. Substituting this into (1) gives,

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

We expect a phase transition in the decision problem when $\kappa \approx 1$. We tested this experimentally using a branch and bound algorithm with the Hungarian heuristic for branching (Carpaneto & Toth 1980). For n=6 to 48, we randomly generated 1000 problems with inter-city distances independently normally distributed with μ=10^6 and σ=10^5. Figure 1 shows the probability that there was a tour less than distance d, plotted against κ. There is a clear phase transition from soluble to insoluble problems that becomes sharper with more cities. Except for problems with 6 cities, there is a critical value of $\kappa = 0.75$ which gives the probability of a tour existing of 0.45 ± 0.04 at all sizes.

Finite size transitions

We can use the constrainedness, κ, to predict the shape as well as the location of phase transitions. Phase transitions in physical systems have been successfully described using finite size scaling methods (Barber 1983). Around a critical temperature T_c, problems of all sizes tend to be indistinguishable except for a change of scale given by a power law in a characteristic length. Here we propose that the constrainedness, κ, plays the role of temperature whilst the problem size, N, plays the role of the characteristic length. This analogy suggests that around some critical constrainedness κ_c, problems

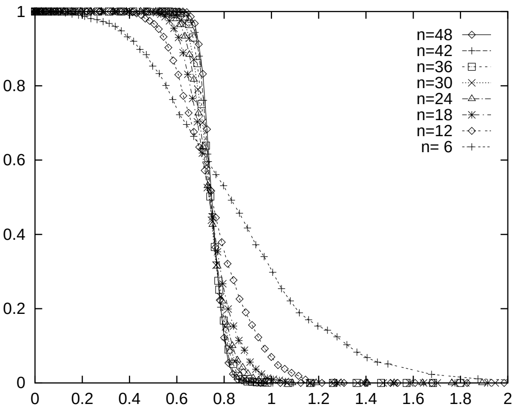

Figure 1. Probability of tour of required length existing in ATSP, plotted against κ for 6 to 48 cities.

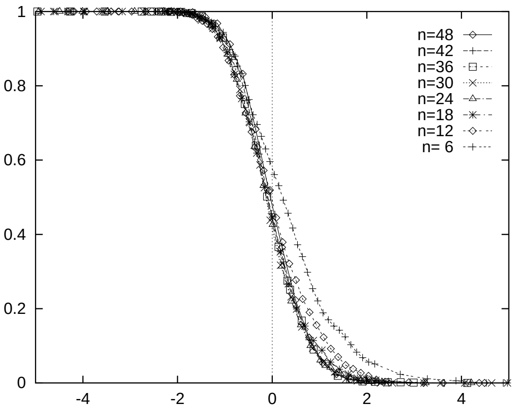

Figure 2. Probability of tour of required length existing in ATSP, against $\frac{\kappa-\kappa_c}{\kappa_c} N^{1/\nu}$ for 6 to 48 cities.

of all sizes will tend to be indistinguishable except for a simple change of scale given by a power law in N. For example, we conjecture that a macroscopic property like the probability of a solution averaged over an ensemble of problems will obey the equation,

$$prob(Sol > 0) = f(\frac{\kappa - \kappa_c}{\kappa_c} N^{1/\nu}) \qquad (4)$$

where f is some fundamental function, $\frac{\kappa-\kappa_c}{\kappa_c}$ is analogous to the reduced temperature $\frac{T-T_c}{T_c}$, and $N^{1/\nu}$ provides the change of scale. Such scaling has been shown to model the probability of a solution in finite size phase transitions in satisfiability (Kirkpatrick & Selman 1994), constraint satisfaction (Gent *et al.* 1995), and number partitioning (Gent & Walsh 1996).

To test this conjecture for the ATSP, in Figure 2 we replot our data against the parameter $\frac{\kappa-\kappa_c}{\kappa_c} N^{1/\nu}$ using $\kappa_c = 0.75$ and $\nu = 2$, both values derived from examination of the data. If (4) holds, the curves will line up when plotted against this rescaled parameter. As predicted, except at $n = 6$, finite size scaling models the probability of a tour existing. A discrepancy at small problem sizes has also been seen in other classes such as satisfiability (Kirkpatrick & Selman 1994) and suggests that finite size scaling provides a very useful but incomplete description of scaling behaviour.

Other macroscopic measures like search cost can often be modelled by finite size scaling (Selman & Kirkpatrick 1996; Gent *et al.* 1995). In Figure 3, we plot the search cost against the rescaled parameter with the same values of κ_c and ν. We use the 90th percentile of the number of nodes searched, as lower percentiles such as median cost were always trivial in that no backtracking occurred. As in many other problem classes, e.g. satisfiability (Mitchell, Selman, & Levesque 1992), search cost displays a distinctive easy-hard-easy pattern through the phase transition.

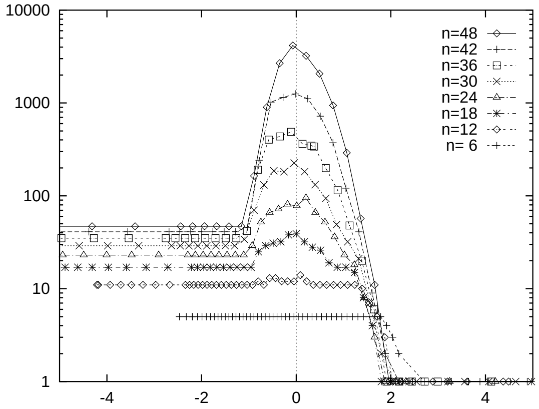

Figure 3. 90th percentile of nodes searched to solve ATSP instances, plotted against $\frac{\kappa-\kappa_c}{\kappa_c} N^{1/\nu}$.

This case study clearly illustrates that our definition of constrainedness is useful in new problem classes. A phase transition occurs, as predicted, at $\kappa \approx 1$. And as expected, by means of finite size scaling we are able to model scaling behaviour of the phase transition.

Constrainedness as a heuristic

Many existing heuristics branch on the most constrained variable, resulting in the least constrained subproblem; i.e. the subproblem with smallest κ. Hence, we propose the heuristic of minimizing κ.

To test this idea, we performed experiments on randomly generated binary CSP's from the class $\langle n, m, p_1, p_2 \rangle$ described earlier. We encoded minimizing κ as a dynamic variable ordering heuristic within the algorithm fc-cbj (i.e. forward checking with conflict-directed backjumping)(Prosser 1993). After instantiating a variable, domain filtering is performed. This may result in a reduction in the size of the domains of future (i.e. uninstantiated) variables and consequently alter the tightness of future constraints (i.e. constraints acting between pairs of future variables). The future sub-problem may then be non-uniform in domain sizes and constraint tightnesses. To measure κ for this reduced problem, we assume it is a representative of the ensemble of problems with the same number of variables, the same domain sizes, and the same number of constraints each of the same tightness as the reduced problem. This is a heuristic assumption which seems to be justified by our results.

When considering a variable v_i as the new current variable we remove it and all constraints involving it from the future sub-problem. We then calculate κ for the future sub-problem using equation (2) and take this as the cost of selecting variable v_i. This is done for all future variables and the variable with minimum cost is selected as the current variable.

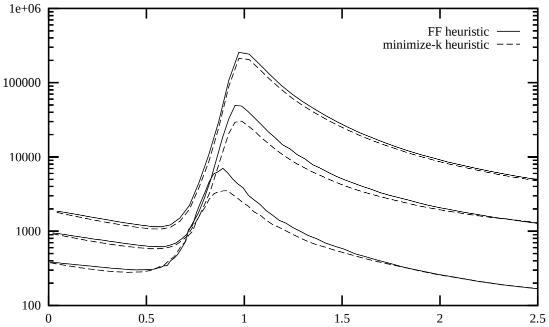

Figure 4. Fail First (FF) and minimize-κ heuristics applied to $\langle 20, 10, p_1 \rangle$ problems using fc-cbj. Mean search effort on y-axis, κ on x-axis. Contours for $p_1 = 1.0$ (top), $p_1 = 0.5$ (middle), $p_1 = 0.2$ (bottom).

We compared the minimize-κ heuristic with an encoding of the fail first (FF) principle (Haralick & Elliott 1980) i.e. selecting the variable with smallest domain. Figure 4 shows the results of experiments performed on $\langle 20, 10, p_1, p_2 \rangle$ problems (i.e. 20 variables, uniform domain size of 10). Constraint density p_1 was varied from 0.2 up to 1.0, for each value of p_1 constraint tightness p_2 was varied, and at each value of p_1 and p_2 1000 problems were generated. The contours shown are for the mean search effort, measured as consistency checks. As can be seen, minimize-κ outperforms the FF heuristic, especially around the phase transition. Although not shown, the same holds for median performance. When search effort is measured as number of trial instantiations of variables, minimize-κ again shows superior mean and median performance. (Gent *et al.* 1996) reports more extensive experiments on the minimize-κ heuristic with similar results. At the peak in search costs, paired-sample t-tests gave values of $t = 12.3$ at $p_1 = 0.2, t = 24.4$ at $p_1 = 0.5$, and $t = 46.3$ at $p_1 = 1.0$, all in favour of minimize-κ. To check the significance of these values we performed an approximate randomization version of the test (Cohen 1995) with a sample of 1000 in each case, which never gave a value above $t = 3.5$. This provides strong statistical evidence that minimize-κ is better than FF in these problem classes.

(Tsang, Borrett, & Kwan 1995) give results on the same problem classes seen in Figure 4, on a range of algorithm/heuristic combinations. For high values of p_1 they report that fc-cbj with the FF heuristic was the best combination studied for problems near the phase transition. That the minimize-κ heuristic can do better is strong evidence that it is a good heuristic. Unfortunately, the complexity of (2) leads to significant overheads in computation, so the heuristic may not give optimal run-times for general CSP solving. We return to this later.

Constrainedness in number partitioning

Many existing heuristics can be justified in terms of minimizing κ. Consider, for example, the Karmarkar-Karp (KK) heuristic for two-way number partitioning (Karmarkar & Karp 1982). The KK heuristic takes a bag B of n numbers to partition and reduces it to a new bag C by removing the largest two numbers x and y, and replacing them by $x - y$ (we assume that $x \geq y$). This commits us to those solutions in which x and y are in opposite partitions.

Let $b = \sum_{i \in B} i$ and $c = \sum_{i \in C} i$. For n numbers drawn uniformly at random from $(0, l]$, setting $m = 2$ in (3) gives the constrainedness $\kappa = \log_2(l)/n$. We will approximate l by $2b/n$, i.e. twice the mean value of the numbers. As $c = b - x - y + (x - y) = b - 2y$, κ goes from $(\log_2(2b/n))/n$ to $(\log_2(2(b - 2y)/n - 1))/(n - 1)$. Since we have no control over n, κ is minimized by maximizing y. Given that $x \geq y$, the maximum y is the second largest element of B. And thus the KK heuristic minimizes κ by picking the two largest elements of B for x and y. Note that we improve the claim of (Karmarkar & Karp 1982) that the motivation behind the KK heuristic is to pick x and y so that $x - y$ is small. The motivation is to pick x and y to minimize the sum of the remaining numbers. Indeed picking x and y so that $x - y$ is minimal (instead of picking x and y maximal) is likely to give very poor performance.

The greedy heuristic for number partitioning (Korf 1995) can also be seen as minimizing κ. The analysis is a little more complex since, unlike the KK heuristic, the greedy heuristic builds partial partitions: it picks the largest number remaining to be partitioned and adds it to the currently smaller partial partition. We will show that this choice is optimal with respect to minimizing κ. We observe that if we have partial partitions with sums s_1 and s_2 and a bag B of numbers remaining to be partitioned then this is equivalent to the problem of partitioning $B \cup \{s_1 - s_2\}$ (we assume that $s_1 \geq s_2$). We minimize κ by maximizing the reduction in the sum of $B \cup \{s_1 - s_2\}$. If we add a number x to the bigger partition, then the sum (and κ) is unchanged. If, however, we add it to the smaller partition then the sum (and κ) decreases. So it is better to put any number in the smaller partition. There are now two cases to consider, depending on whether $x > s_1 - s_2$

or $x \leq s_1 - s_2$. First suppose that $x > s_1 - s_2$. The sum reduces by $x + s_1 - s_2 - ((x + s_2) - s_1) = 2(s_1 - s_2)$. If, however, $x \leq s_1 - s_2$, the sum reduces by $x + s_1 - s_2 - (s_1 - (x + s_2)) = 2x \leq 2(s_1 - s_2)$. The first difference is always the larger and is therefore preferred. To minimize κ a heuristic should pick a x in B such that $x \geq s_1 - s_2$, or failing that the largest x in B. The greedy heuristic achieves this by picking the largest x in B and putting it in the smaller partition.

Although both the greedy and KK heuristics minimize κ, the KK heuristic appears to perform significantly better (Korf 1995; Gent & Walsh 1996). This is due to the different methods used by the two heuristics in decomposing problems into subproblems. For its method, each heuristic does as well as possible with respect to minimizing κ. Comparisons of the changes in κ between the two methods of decomposition may offer an explanation of the superiority of KK over greedy.

Not all proposed heuristics minimize κ: for example (Horowitz & Sahni 1974) consider partitioning numbers in increasing order. However, constrainedness suggests considering numbers in decreasing order. This was shown to be superior in (Rubin 1976).

Proxies for constrainedness

Although minimizing κ appears to be a good heuristic, it can be expensive to compute according to formula (2), as in the CSP experiments discussed earlier. We may therefore use a proxy which is cheaper to compute.

If we assume that all constraints in a problem have the same tightness, and that each variable is in the same number of constraints, we can ignore the numerator of (2) as it will be the same whichever variable we instantiate. The variable chosen should then be the one that maximizes the denominator of (2), and is equivalent to instantiating the variable with smallest domain. This is the fail-first (FF) heuristic (Haralick & Elliott 1980).

An alternative assumption is that all variables have the same domain size. This is valid if all variables have identical domain sizes and we use a backward checking algorithm, i.e. an algorithm that does not perform domain filtering of the future variables. The denominator will now be the same whichever variable we instantiate. If we further assume that all constraint tightnesses are the same, the numerator becomes the cardinality of the set of constraints acting between future variables and between future and past variables. We minimize the numerator of (2) by choosing a variable that has most constraints with past variables. This corresponds to the maximum cardinality heuristic described in (Dechter & Meiri 1994).

We may take advantage of both numerator and denominator of (2). One way to do this is to choose the variable with smallest domain size (maximizing the denominator) and break ties by choosing the tied variable in most constraints (minimizing the numerator, assuming uniform constraint tightness). This is the Brelaz heuristic (Brelaz 1979).

This analysis has identified three state of the art heuristics for CSP. Domain knowledge may still be needed to convert the idea of minimizing κ into a heuristic with low overheads. However, by considering how to minimize κ, we can remove much of the intuition involved in developing heuristics for a new domain. While intuition is valuable, it can often be misleading or even wrong, as we saw in discussing heuristics for number partitioning. Furthermore, intuition about new domains can be hard to achieve. We therefore see this reduction in the role of intuition in heuristic design as a significant contribution.

Related work

(Smith 1995) proposed a heuristic that simply maximizes the expected number of solutions, $\langle Sol \rangle$. Given a choice of two subproblems with equal $\langle Sol \rangle$, the heuristic of minimizing κ will branch into the smaller problem in the expectation that this is less constrained. Initial experiments have failed to show which heuristic, if either, is better (Gent *et al.* 1996).

(Hooker & Vinay 1995) investigate the Jeroslow-Wang heuristic for satisfiability. They propose the "satisfaction hypothesis", that it is best to branch into subproblems that are more likely to be satisfiable, but reject this in favour of the "simplification hypothesis", that it is best to branch into simpler subproblems with fewer and shorter clauses after unit propagation. Minimizing κ is related but not identical to both these hypotheses: in general it will seek out simple problems that are likely to be soluble.

(Musick & Russell 1992) model search using an abstracted Markov process. They identify regions where problems are easy and outside which it is very hard to find a solution. It would be fruitful to explore the connections between constrainedness, and the transition probabilities of such Markov processes.

(Gent & Walsh 1996) suggest that we use κ to compare heuristics. For example, in number partitioning the KK heuristic almost always returns the optimal and exact partition when $\kappa < 0.4$, but the greedy heuristic only performs well for $\kappa < 0.2$.

Phase transitions have also been observed in problems based on real data (Gent & Walsh 1995a). The constrainedness of a problem depends on the ensemble from which it is drawn. We may not know the ensemble from which a real problem is drawn, so naive measurements of κ may mislead us. The role of problem representation must also be taken into account, as in a study such as (Borrett & Tsang 1995). Further work in this area is vital if this research is to be of value in understanding and solving real problems.

Conclusions

We have defined a very general parameter κ, pronounced 'kappa', that measures the constrainedness of an ensemble of combinatorial problems. This generalises and unifies many parameters previously used to study phase transitions in NP-complete problem classes, and allows the rapid identification of phase transitions in new problem domains. It also allows the direct comparison of transitions in previously incomparable classes.

Constrainedness is also useful as a heuristic to guide search. Many existing heuristics can be seen as minimizing κ or proxies for κ. This offers a unified understanding of many widely disparate heuristics.

References

Abramowitz, M., and Stegun, I., eds. *Handbook of mathematical functions.* Dover.

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

Borrett, J and Tsang, E 1995. On the selection of constraint satisfaction formulations. Tech. report CSM-254, Dept. of Computer Science, Essex University.

Brelaz, D. 1979. New methods to color the vertices of a graph. *Comms. ACM* 22:251–256.

Carpaneto, G. and Toth, P. 1980. Some 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.

Cohen, P. 1995. *Empirical methods for Artificial Intelligence.* MIT Press.

Crawford, J., and Auton, L. 1993. Experimental results on the crossover point in satisfiability problems. In *Proceedings of AAAI-93*, 21–27.

Dechter, R., and Meiri, I. 1994. Experimental evaluation of preprocessing algorithms for constraint satisfaction problems. *Artificial Intelligence* 68:211–242.

Gent, I., and Walsh, T. 1995a. Phase transitions from real computational problems. In *Proceedings of the 8th International Symposium on AI*, 356–364.

Gent, I., and Walsh, T. 1995b. The TSP phase transition. Research report 95/178, Department of Computer Science, University of Strathclyde.

Gent, I., and Walsh, T. 1996. Phase transitions and annealed theories: Number partitioning as a case study. In *Proceedings of ECAI-96.*

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.; Smith, B.; and Walsh, T. 1996. An empirical study of dynamic variable ordering heuristics for the constraint satisfaction problem. Research Report 96.05, School of Computer Studies, University of Leeds.

Haralick, R., and Elliott, G. 1980. Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence* 14:263–313.

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

Hooker, J., and Vinay, V. 1995. Branching Rules for Satisfiability. *Journal of Automated Reasoning* 15:359–383.

Horowitz, E., and Sahni, S. 1974. Computing partitions with applications to the knapsack problem. *J. ACM* 21(2):277–292.

Karmarkar, N., and Karp, R. 1982. The difference method of set partitioning. Technical Report UCB/CSD 81/113, Computer Science Division, University of California at Berkeley.

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

Korf, R. 1995. From approximate to optimal solutions: A case study of number partitioning. In *Proceedings of the 14th IJCAI*, 266–272.

Mitchell, D.; Selman, B.; and Levesque, H. 1992. Hard and easy distributions of SAT problems. In *Proceedings of AAAI-92*, 459–465.

Musick, R., and Russell, S. 1992. How long will it take? In *Proceedings of AAAI-92*, 466–471.

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* 81:127–154.

Rubin, F. 1976. Partition of integers. *ACM Transactions on Mathematical Software* 2(4):364–374.

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

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

Smith, B. 1995. Re: variable choice. *CSP List Digest* 75. Mailing list archived at ftp.cs.city.ac.uk in /pub/constraints/archive/csp-list/95.10.75.gz.

Tsang, E.; Borrett, J.; and Kwan, A. 1995. An attempt to map the performance of a range of algorithm and heuristic combinations. In *Hybrid Problems, Hybrid Solutions*, 203–216. IOS Press.

Williams, C., and Hogg, T. 1994. Exploiting the deep structure of constraint problems. *Artificial Intelligence* 70:73–117.

Zhang, W., and Korf, R. 1992. An average-case analysis of branch-and-bound with applications: Summary of results. In *Proceedings of AAAI-92*, 769–775.