ELSEVIER
Theoretical Computer Science 265 (2001) 265–283
Theoretical Computer Science
www.elsevier.com/locate/tcs

Constructing an asymptotic phase transition in random binary constraint satisfaction problems

Barbara M. Smith *

School of Computing & Mathematics, University of Huddersfield, Huddersfield HD1 3DH, UK

Abstract

The standard models used to generate random binary constraint satisfaction problems are described. At the problem sizes studied experimentally, a phase transition is seen as the constraint tightness is varied. However, Achlioptas et al. showed that if the problem size (number of variables) increases while the remaining parameters are kept constant, asymptotically almost all instances are unsatisfiable. In this paper, an alternative scheme for one of the standard models is proposed in which both the number of values in each variable's domain and the average degree of the constraint graph are increased with problem size. It is shown that with this scheme there is asymptotically a range of values of the constraint tightness in which instances are trivially satisfiable with probability at least 0.5 and a range in which instances are almost all unsatisfiable; hence there is a crossover point at some value of the constraint tightness between these two ranges. This scheme is compared to a similar scheme due to Xu and Li.

Keywords: Constraint satisfaction; Random problems; Phase transitions

1. Introduction

Following the paper by Cheeseman et al. [3] in 1991, phase transition phenomena in NP-complete problems, including constraint satisfaction problems (CSPs), have been widely studied. In CSPs, using randomly generated instances, a phase transition is seen as the tightness of the constraints is varied; when the constraints are loose, there are many solutions and it is easy to find one; when the constraints are tight, it is easy to prove that there are no solutions. Hard instances occur over the range of values of the constraint tightness corresponding to an abrupt fall in the probability that an instance has a solution from near 1 to near 0. The peak in average cost of solving an instance

* This work was completed while the author was employed at the University of Leeds.
E-mail address: b.m.smith@huc.ac.uk (B.M. Smith).

PII: S0304-3975(01)00166-9

occurs, empirically, at or near the *crossover point*, where the probability of a solution is 0.5.

A number of possible models for generating random CSPs are discussed in Section 2. However, Achlioptas et al. [2] showed that these standard models do not have a phase transition in the same sense as other NP-complete problems such as graph colouring, because 'asymptotically, almost all instances generated will not have a solution'. They proposed an alternative model for generating random instances, described below. However, in this paper, it is shown that the difficulty is not necessarily with the way in which the standard models generate instances, but in the way that the parameters change as the problem size increase, and it is proved that for one of the standard models, there will asymptotically be a crossover point when the constraint tightness is within a certain range.

2. Random binary CSPs

A CSP consists of a set of variables $\mathrm{X} = \{x_1, \dots, x_n\}$; for each variable, a finite set D_i of possible values (its domain); and a set of constraints, each of which consists of a subset $\{x_i, \dots, x_j\}$ of X and a relation $\mathrm{R} \subseteq \mathrm{D}_i \times \cdots \times \mathrm{D}_j$; informally, the constraint specifies the allowed tuples of values for the variables it constrains. A solution to a CSP is an assignment of a value from its domain to every variable such that all the constraints are satisfied; a constraint is satisfied if the tuple of values assigned to the variables it constrains is in the constraint relation. A k-ary constraint constrains k of the problem variables; in a binary CSP, all the constraints are binary.

Instances of binary CSPs can easily be generated randomly since a binary CSP can be represented by a constraint graph and a set of constraint matrices. The constraint graph of a binary CSP is a graph in which there is a node representing each variable and for every constraint there is an edge linking the affected variables. A constraint relation between a pair of variables with m_1 and m_2 values in their respective domains can be represented by an $m_1 \times m_2$ matrix of Boolean values, in which a 0 indicates that the constraint does not allow the simultaneous assignment of the corresponding pair of values, and 1 that it does. Each entry of 0 in a constraint matrix corresponds to a *nogood*, i.e. a forbidden pair of variable-value assignments. Random binary CSPs can be generated using a simple model described by the tuple $\langle n, m, p_1, p_2 \rangle$, where n is the number of variables, m is the uniform domain size, p_1 is a measure of the density of the constraint graph, and p_2 is a measure of the tightness of the constraints. We first generate a constraint graph G, and then for each edge in this graph, we choose pairs of incompatible values for the associated conflict matrix. (Note that each edge will have a different constraint matrix associated with it.) There are several variants of this basic model, differing in how they treat p_1 and p_2; we can either choose exactly $p_1 n(n-1)/2$ of the available edges or choose each edge independently with probability p_1. Similarly, in forming a constraint matrix, we either choose exactly $p_2 m^2$ of the possible pairs of values to be the incompatible pairs for this constraint, or choose each pair independently with probability p_2.

If both p_1 and p_2 are treated as probabilities, we get the model termed Model A in [16]; if both p_1 and p_2 are proportions, we get Model B. Many experimental and theoretical studies of random binary CSPs have used a model equivalent to one of these (see [7] or [13] for a survey). Sometimes, p_1 is treated as a proportion and p_2 as a probability, giving Model D, and for completeness, Model C has p_1 as a probability and p_2 as a proportion. In Models B and D, the constraint density, p_1, cannot vary continuously, since the number of constraints must be integral, and the number of constraints might be a better parameter for these models. Some researchers have indeed used models equivalent to Models B or D, using the number of constraints as a parameter.

3. Phase transition studies in binary CSPs

Most investigations of phase transitions in binary CSPs have been more concerned with observing and explaining the behaviour of samples of problems at specific parameter values than with how behaviour changes with problem size. Some have been concerned with predicting where the hardness peak will occur (e.g. [16]). Others consider the rare instances in the under-constrained region whose solution cost is extremely high (e.g. [12, 8, 17]). It has also become standard practice for CSP researchers proposing new algorithms or heuristics to show performance comparisons across the phase transition, both because performance improvements in the hard region are most worthwhile, and because one algorithm may not be uniformly better or worse than another across the whole range of constraint tightnesses.

Furthermore, most experimental studies have not considered a wide range of problem sizes, because of the difficulty of solving large samples of CSPs with many variables. Typically, random CSPs have been generated with $m = 5$ or 10, and problem sizes have then rarely been more than 100 variables. Occasionally problems with $m = 3$ have been used, e.g. [6], and then larger problem sizes have been possible. Prosser [15] carried out an extensive investigation of CSPs generated using Model B. In three series of experiments, he showed the effect of varying one of the other parameters as well as p_2; hence, in one series, the number of variables varied (from 20 to 60) while m and p_1 were constant.

Grant and Smith [9, 10], in experiments comparing two CSP algorithms, explicitly considered the effect of increasing problem size, and proposed keeping the average degree of the constraint graph constant at the same time. Empirically, over the range of sizes that were considered (20–50) this ensures that the peak in average difficulty occurs at roughly the same value of p_2 for each value of n. There is some theoretical justification for this. The crossover point is predicted to occur when the expected number of solutions, E(N) is equal to 1. Since $E(N) = m^n(1 - p_2)^{nd/2}$, if m and d are fixed as n increases, the value of p_2 for which $E(N) = 1$ will also be constant.

However, this has been shown to be a poor predictor of the crossover point when the constraint graph is sparse. In [7], it was shown that, if the degree of the constraint graph rather than the constraint density is kept constant, the difficulties identified by Achlioptas et al. emerge much more slowly as problem size increases. Nevertheless, their asymptotic result still holds, so that the crossover point will not occur at the same value of p_2 indefinitely as n increases.

Smith and Dyer [16] considered two ways in which the other parameters of Model B might change as n increases (p_1 constant; m constant or $m = n$) in considering the accuracy of the prediction of the crossover point given by $\mathrm{E(N)} = 1$. They pointed out that if m is constant the crossover point will eventually be smaller than the smallest value of p_2 allowed by Model B, for large enough n, so that almost all instances generated will have no solution. This will also be true if $m = n$.

To summarise, there has been little discussion of how the parameters m and p_1 should increase with n, largely because the focus has been on discussing the behaviour of CSPs at experimental sizes, rather than on asymptotic behaviour. There has, however, been an implicit assumption that as problem size increases, the domain size, m, should remain constant. This is perhaps by analogy with graph colouring problems. Graph colouring problems can easily be represented as binary CSPs, with a variable for each node in the graph; the domain is the set of available colours. 3- and 4-colouring problems were considered in [3]. Clearly, the domain size for k-colouring problems is k, however large the graphics.

Achlioptas et al. [2] showed that these standard models do not have a phase transition in the same sense as other NP-complete problems such as graph colouring, because asymptotically, as $n \to \infty$, almost all instances are unsatisfiable, provided that $p_2 > 0$ if p_2 is a probability and $p_2 \geqslant 1/m^2$ if it is a proportion. They show, in fact, that asymptotically almost all instances are *trivially* unsatisfiable, in the sense that there will be at least one variable which has no value consistent with the values of adjacent variables. This is shown to be true if the average degree of the nodes in the constraint graph is constant; if the constraint density is constant, then the number of constraints grows even faster with n so that trivially unsatisfiable instances will occur earlier. In both cases the domain size is assumed to be constant with n. They attribute the difficulty to the fact that the constraint matrices are random rather than structured (as for instance in graph colouring) and suggest that it can be dealt with by changing the way in which the constraint matrices are generated.

In [2], Model E is proposed as an alternative: rather than generating the constraint graph and the constraint matrices separately, in Model E, a prescribed number of nogoods are selected at random. However, a disadvantage of this model is that as the number of nogoods increases it rapidly generates a complete constraint graph, most of the constraints having only a small number of forbidden pairs of values. This makes it unrepresentative of the CSPs that arise in modelling real problems.

Models A, B, C and D specify only how a set of instances should be generated, given values of n, m, p_1 and p_2. They say nothing about how the parameters should change as n increases. It has been shown that if m and either p_1 or the average degree of the constraint graph, d, are constant as n increases, there is not an asymptotic phase transition. However, this does not necessarily require us to use a different set of models. We could instead consider changing m and/or p_1 or d as n increases. Here, a scheme for using Model D and changing the other parameters with n is proposed. It is shown that this guarantees that the phase transition will occur within a specified range of values of p_2, whatever the value of n. Since Model D is otherwise unchanged, this modification does not require any change to the way in which the constraint matrices are generated, unlike Model E.

4. Backtrack-free search

A CSP instance can be solved, or it can be proved that there is no solution, by a complete search algorithm, for instance a simple backtracking algorithm (BT). This algorithm attempts to build up a consistent solution to the CSP by considering each variable in turn, and trying to assign a value to it which is consistent with the existing assignments. If a consistent value is found, the next variable is tried; otherwise, BT backtracks to the most recently assigned variable which still has an untried value, and tries a different value for it. The algorithm proceeds in this fashion until either a complete solution is found or every possible value for the first variable has been tried without success, in which case the problem has no solution.

A transition from satisfiable to unsatisfiable instances has been observed in random binary CSPs if, for constant n, m and p_1, a large number of $\langle n, m, p_1, p_2 \rangle$ instances are generated at each of a range of values of p_2 and each instance is solved using an algorithm such as BT. Fig. 1 shows the typical pattern seen if the average search cost (in this case, the median) is computed for each value of p_2. Here, the number of consistency checks, i.e. the number of references to an element of a constraint matrix, is used as the measure of search cost. Similar behaviour has been seen for a wide range of parameter values and complete search algorithms, as shown for instance in [15].

In Fig. 1, the instances were generated using Model B, but there is little difference in the median cost curves between Models A, B, C and D. The greatest median cost occurs where approximately half of the generated instances have solutions and half do not, i.e. around the crossover point, where the probability that a random instance has a solution is 0.5. There is a narrow region around the peak where the generated populations contain a mixture of satisfiable and unsatisfiable instances; for smaller values of p_2, all the instances can be solved, and are increasingly easy to solve as the constraints become looser; for larger values of p_2, none of the instances have solutions, and as the constraints become tighter, they are increasingly easy to prove unsatisfiable.

In the 'easy-satisfiable' region, Fig. 1 shows a point where the gradient of the median curve changes, at $p_2 = 0.09$ approximately. This is also a kind of crossover point: it is the point where the algorithm can solve 50% of the generated instances without backtracking. In other words, in 50% of instances, as each variable is considered, there

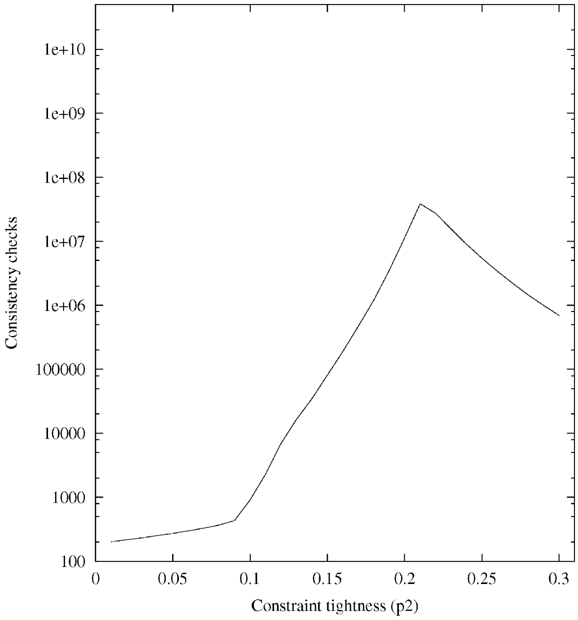

Fig. 1. Median cost of solving Model B problems with $n = 20$, $m = 10$, $p_1 = 1.0$ using BT—10,000 problems per p_2.

is a value consistent with the past assignments, and the algorithm never has to return to a previous variable. Unlike the solubility crossover point, this is algorithm dependent. More efficient algorithms than BT, such as forward checking [11], which considers the effect of each assignment on the variables not yet assigned, can avoid failures due to incorrect choices which BT cannot, and so solve instances without backtracking at larger values of p_2.

The region where most problems can be solved without backtracking by an algorithm is a region of 'trivial satisfiability'. This might be viewed as roughly complementary to the region of trivial unsatisfiability identified by Achlioptas et al. Clearly, if an instance can be solved without backtracking, it has a solution. Thus, the probability that a random instance can be solved without backtracking is a lower bound on the probability that it has a solution, and the constraint tightness at which the probability of backtrack-free search is 0.5 is a lower bound on the crossover point. Fig. 1 suggests that for BT this bound will not be very tight. However, it has the advantage that it can be calculated, as will be shown below. If we can generate instances in such a way that the probability of backtrack-free search is always positive for some range of values of p_2, for all values of n, we shall no longer have the situation where almost all problems become trivially unsatisfiable for sufficiently large values of n. This would then be the first step towards a CSP model which gives an interesting asymptotic phase transition.

For Model D, the probability that BT can solve a problem without backtracking can be calculated. This is given in [17] for complete constraint graphs. For incomplete constraint graphs, if the variables are instantiated in the order $1, 2, \ldots, n$, the probability that there is at least one value in the domain of variable i which is consistent with the assignment already made is:

$$1 - (1 - q_2^{d_i})^m,$$

where d_i is the past degree of variable i, i.e., the number of variables in $1, 2, \ldots, i-1$ which constrain it. This is so because the nogoods in the constraint matrices are generated independently. The probability that at least one value of every variable is consistent with the previous assignments is

$$\prod_{i=2}^{n} (1 - (1 - q_2^{d_i})^m).$$

Let w be the maximum past degree of any variable in this ordering, i.e. the *width* of the ordering. Then

$$\prod_{i=2}^{n} (1 - (1 - q_2^{d_i})^m) \geqslant \prod_{i=2}^{n} (1 - (1 - q_2^{w})^m) = (1 - (1 - q_2^{w})^m)^{n-1}.$$

There is a well-known algorithm for finding a ordering of the nodes with minimum width [5], and Dyer and Frieze have shown that the minimum width of a random graph is almost surely bounded by the average degree of the graph. [1] Experimental results suggest that in randomly generated graphs, even of small size, the minimum width will be strictly less than the average degree, unless the graph is complete (in which case, of course, the width of any ordering is equal to the degree) or extremely sparse.

We can therefore use $(1 - (1 - q_2^d)^m)^{n-1}$, where d is the average degree, as a lower bound on the probability of backtrack-free search using BT. If we can ensure that the value of this is at least 0.5 for some fixed value of the constraint tightness, for all n, we know that the true probability of backtrack-free search at that constraint tightness is at least 0.5, and hence we have the required region of trivial solubility.

How do we keep $(1 - (1 - q_2^d)^m)^{n-1}$ constant at a specified value of p_2, when n is increasing? To achieve this, $(1 - q_2^d)^m$ must decrease, and hence either m must increase or d must decrease. Intuitively, if the number of variables, n, increases while the other parameters are kept constant, it becomes harder to find a value for every variable satisfying the constraints between it and the past assignments. This can be compensated for by increasing the probability that an individual variable is consistent with the past assignments, either by increasing the variable's domain size, or by decreasing the number of past variables which constrain it. Decreasing the average degree of the constraint graph as n increases is not a desirable option, as it will eventually result in the graph being disconnected, so m must increase.

[1] Personal communication: this is a consequence of results in [14].

5. Upper bound on the crossover point

As shown in the last section, a lower bound on the crossover point is given by the constraint tightness, $\tilde{p}_2$, for which the probability of backtrack-free search using BT is at least 0.5. An upper bound, $\hat{p}_2$, is given by the value for which the expected number of solutions, E(N), is 0.5; for any $p_2 > \hat{p}_2$, the probability that there is a solution is at most 0.5.

For the standard models for random binary CSPs, $\mathrm{E(N)} = m^n(1 - p_2)^{nd/2}$. If we consider the value of p_2 at which $\mathrm{E(N)} = 1$, we must have $m(1 - p_2)^{d/2} = 1$. It is clear that if m is increasing and d is constant, then to satisfy this equation, p_2 must also increase; as $m \to \infty$, $p_2 \to 1$. Hence, the upper bound on the crossover point, $\hat{p}_2$, will also increase and we shall therefore not be able to show that there is a range of values of the constraint tightness for which instances are unsatisfiable, with high probability. To ensure that $\hat{p}_2$ does not increase with m, d must also increase. However, bearing in mind the intuition given at the end of the last section, d must not increase so fast that the region of trivial solubility disappears; increasing d and n makes instances more likely to have no solution, whereas increasing m makes them more likely to have a solution, and a balance must be struck.

There is already evidence, in fact, that increasing m and d arbitrarily with n will not necessarily lead to an asymptotic phase transition. In [16], the class of problems $\langle n, n, 1, p_2 \rangle$ was studied, i.e. problems in which $m = n$ and the constraint graph is complete. For these problems, the crossover point $\to 0$ as $n \to \infty$. Although the number of values for each variable is increasing, the degree of the constraint graph is also increasing ($d = n - 1$), and too fast to allow instances to remain satisfiable.

6. A proposed scheme

To confine the crossover point between values of p_2 which are either constant or converging as n increases, both m and d must increase with n. In order to keep as close as possible to the experimental use of existing models, in which m and p_1 or d are fixed, it is desirable to increase d and m only slowly with n.

Suppose we have a base population of problems, defined by the parameters n_0, m_0, d_0, which will give the initial value of $\tilde{p}_2$, say $\tilde{\pi}_2$. To increase d slowly with n, suppose it grows as $\log n$, e.g. $d = a \log n + c$, where a is a constant, and c is determined by n_0, d_0 and a. Since almost every random graph with $(a/2) n \log n$ edges is connected if $a > 1$, we should choose $a > 1$. If the constraint graph is disconnected, each component of the graph forms a CSP, and these would in practice be solved separately, so that in generating random instances it is a realistic requirement that the constraint graph should be connected.

m is defined so that for any value of n, the lower bound on the crossover point is $\tilde{\pi}_2$, i.e. m satisfies $(1 - (1 - (1 - \tilde{\pi}_2)^d)^m)^{n-1} = 0.5$, where n and d are related as just described. This ensures that, for all n, instances with $p_2 \leqslant \tilde{\pi}_2$ are trivially satisfiable,

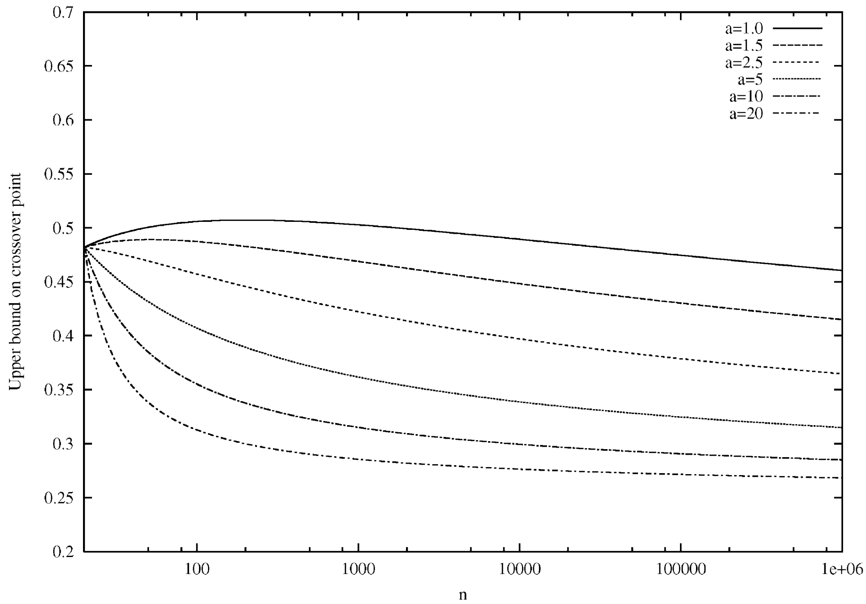

Fig. 2. Calculated upper bound on crossover point, from a base population with $n=20$, $m=5$, $d=5$.

with probability at least 0.5. To ensure an asymptotic phase transition, the upper bound on the crossover point, $\hat{p}_2$, should have a limiting value as $n\to\infty$ which is strictly <1. This will ensure that there is, asymptotically, a region where instances have no solution, for all n. However, since m and d are now defined in terms of n, nothing further can be done to ensure the required behaviour; it only remains to be seen whether the instances generated do in fact behave in this way.

7. Asymptotic behaviour

First, calculations using different base populations and different values of a suggest that increasing m and d with n as proposed does give the required behaviour. Fig. 2 shows how $\hat{p}_2$ varies with n, for a base population with $n=20$, $m=5$ and $d=5$, for a range of values of a. If a is small, i.e. d is increasing extremely slowly with n, $\hat{p}_2$ increases initially with n. However, Fig. 2 suggests that it does eventually decrease, so that there is a range of values of p_2 for which unsatisfiable instances occur. By construction, the lower bound on the crossover point, given by $\tilde{\pi}_2$, is constant for all n, and hence the probability that an instance has a solution, p_{sat}, is at least 0.5 for $p_2<\tilde{\pi}_2$, for all n. The empirical evidence therefore suggests that there is, asymptotically, a phase transition somewhere between $\tilde{\pi}_2$ and the limiting value of $\hat{p}_2$, which is <1.

If this scheme were to be used to generate Model D instances for small values of n, there are some practical considerations which would need to be taken into account.

Specifically, m must be an integer, and nd must be an even integer, since the number of constraints is $nd/2$. Having calculated d from $d = a\log n + c$, it might then be necessary to adjust it slightly. Then, rather than choosing m as above, a possible choice is the smallest integer such that $(1-(1-(1-\tilde{\pi}_2)^d)^m)^{n-1} > 0.5$. The upper bound on the crossover point would not then follow exactly the curves shown in Fig. 2, when n and a are such that m is small, and the lower bound on the crossover point would not be constant at $\tilde{\pi}_2$. However, since the principal concern here is with asymptotic behaviour, these variations when the values are small do not significantly affect the argument.

To show that the proposed scheme does give the required asymptotic behaviour, we need to show what happens to the upper bound on the crossover point. Suppose that $\hat{p}_2$ is the constraint tightness for which the expected number of solutions is ε, for $0 < \varepsilon < 1$. Then for $p_2 > \hat{p}_2$, $p_{\mathrm{sat}} < \varepsilon$.

$$\begin{aligned} 1 - \hat{p}_2 &= \varepsilon^{2/nd} m^{-2/d} \\ &\to m^{-2/d} \quad \text{as } m, d \to \infty \text{ with } n. \end{aligned}$$

Hence, $\log \hat{q}_2 \to -2\log m/d \to -\infty/\infty$ as $m, d \to \infty$ with n, where m is given by

$$(1-(1-(1-\tilde{\pi}_2)^d)^m)^{n-1} = 0.5$$

and d is given by

$$d = a\log n + c.$$

It can be shown (see Appendix A.1) that

$$\lim_{n\to\infty} \hat{p}_2 = 1-(1-\tilde{\pi}_2)^2.$$

For instance, for the problems shown in Fig. 2, $\tilde{\pi}_2 = 0.1343$ and $\hat{p}_2 \to 0.2506$.

Since ε can be arbitrarily close to 0, we have

$$\lim_{n\to\infty} p_{\mathrm{sat}} = 0 \quad \text{for } p_2 > 1-(1-\tilde{\pi}_2)^2.$$

Note that the limiting value of $\hat{p}_2$ does not depend on a, although Fig. 2 shows that the value of a does affect the rate of convergence.

By increasing m and d with n as specified, we can therefore ensure that for all n, $p_{\mathrm{sat}} > 0.5$ for $p_2 < \tilde{\pi}_2$, the value given by the base population, and $\lim_{n\to\infty} p_{\mathrm{sat}} = 0$ for $p_2 > 1-(1-\tilde{\pi}_2)^2$. Hence, asymptotically there is a crossover point between these two values of the constraint tightness, and we do have an asymptotic phase transition with this scheme.

Furthermore, both d and m can increase quite slowly with n, so that for the range of problem sizes that are likely to be solvable in practice, this scheme will be close to schemes in which m and d are constant. This depends on the value of a; if a is

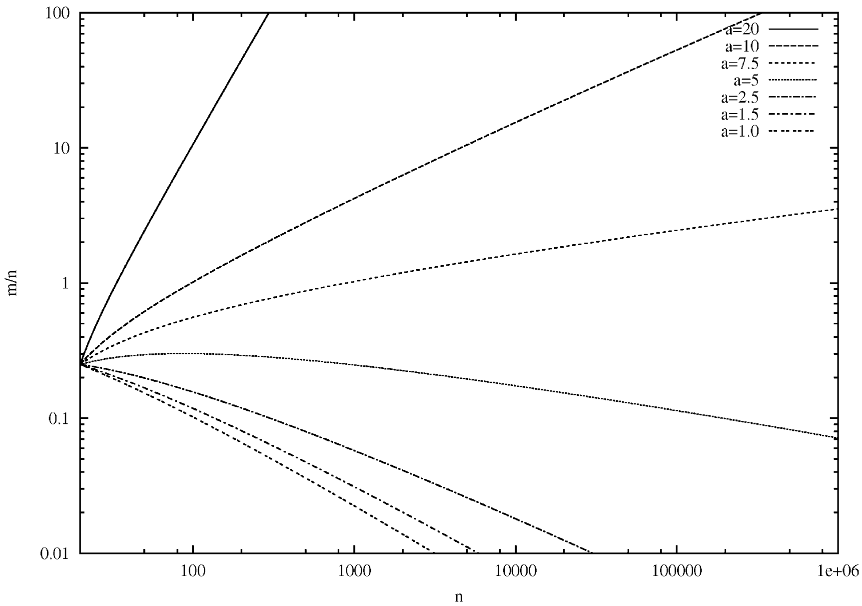

Fig. 3. Ratio of m to n, from a base population with $n=20$, $m=5$, $d=5$.

large, d increases relatively rapidly with n, and m also increases very fast, and will become larger than n. Fig. 3 shows that for the base population shown in Fig. 2, m becomes much larger than n, and appears to increase without limit, if $a \geqslant 7.5$. m remains smaller than n for $a \leqslant 5$, and m/n rapidly approaches 0 for $a \leqslant 2.5$.

It can be shown (Appendix A.2) that

$$\lim_{n\to\infty} m/n = \infty \quad \text{if } a\log(1-\tilde{\pi}_2)+1 \leqslant 0$$
$$= 0 \quad \text{otherwise}$$

For the populations shown in Fig. 2, $\tilde{\pi}_2 = 0.1343$, so that $\lim_{n\to\infty} m/n = \infty$ for $a \geqslant 6.934$, which accords with Fig. 3.

Fig. 2 suggests that choosing a large value of a will give more rapid convergence of the upper bound on the crossover point. On the other hand, m will then grow much faster than n, giving problems very different from the instances normally generated in experimental studies. For the initial parameters used in Figs. 2 and 3, a value of a around 2.5 would give an upper bound on the crossover point which is always less than its initial value, and a rapid decrease in the ratio m/n. Hence, such a scheme has a guaranteed asymptotic crossover point but is not too dissimilar at experimental sizes from a scheme with m and d constant.

8. Related work

A similar scheme was proposed independently by Xu and Li [19] for Model B. Their scheme also applies to non-binary constraints. In the binary case, the constraint graph has $(a/2)n\log n$ edges, using the notation of this paper. The number of values in each variable's domain is n^γ, for constant $\gamma > 0$.

The constraint tightness, $p_{2\text{crit}}$, at which $\mathrm{E(N)} = 1$ has been used as a predictor of the crossover point [18, 16]. In Xu and Li's scheme, $p_{2\text{crit}} = 1 - \mathrm{e}^{-2\gamma/a}$. They show, by considering the second moment, $\mathrm{E(N^2)}$, that if $\gamma > 1/2$ and $\mathrm{e}^{-2\gamma/a} \geqslant 1/2$ then

$$\begin{aligned}\lim_{n\to\infty} p_{\text{sat}} &= 1 \quad \text{for } p_2 < p_{2\text{crit}}\\ &= 0 \quad \text{for } p_2 > p_{2\text{crit}}.\end{aligned}$$

Since $\gamma > 1/2$ and $\mathrm{e}^{-2\gamma/a} \geqslant 1/2$, $a \geqslant 2\gamma/\log 2 > 1/\log 2 = 1.4427$. Hence there is a range of acceptable values of a between 1 and 1.4427, for which the constraint graphs are connected, and similarly a range of values of γ between 0 and 1/2, where Xu and Li's result does not apply, as far as the lower limit on the crossover point is concerned. It is still true, however, that $\lim_{n\to\infty} p_{\text{sat}} = 0$ for $p_2 > 1 - \mathrm{e}^{-2\gamma/a}$.

If we assume Model D rather than Model B, we can use the probability of backtrack-free search, as in Section 4, whether or not a and γ have values in the ranges to which Xu and Li's result applies. Hence, it can be shown that there must still be a crossover point asymptotically, even if we cannot prove its precise location. It can be shown (see Appendix A.3) that

$$\lim_{n\to\infty} p_{\text{sat}} = 1 \quad \text{if } p_2 < 1 - \mathrm{e}^{-\gamma/a}$$

and hence there is a crossover point, for p_2 between $1 - \mathrm{e}^{-\gamma/a}$ and $1 - \mathrm{e}^{-2\gamma/a}$, even when Xu and Li's result does not apply. It is worth noting that the relationship between the upper and lower bounds on the crossover point is the same as found previously in Section 7.

Models B and D differ only in the generation of the constraint matrices; p_2 is the proportion of nogoods in each matrix in Model B and the probability that a pair of values is nogood in Model D. Since a Model B population is more homogeneous than a Model D population, it seems likely that if Model D has an asymptotic crossover point then so does Model B. However, this needs to be confirmed before it can be assumed that the result just given applies to Xu and Li's scheme. On the other hand, it was shown in [16] that Models A and B are not equivalent asymptotically, so the results of this paper may not transfer to all the standard models.

9. Conclusions

It has been shown that it is possible to ensure an asymptotic crossover point in one of the standard random binary CSP models, Model D, by increasing other parameters of the model with the number of variables. This counters the objections raised by Achlioptas et al. to these models, that asymptotically almost all instances are trivially unsatisfiable. Hence, it is not necessary to change the way in which the constraint matrices are generated in order to produce asymptotically interesting behaviour. However, since there is an asymptotic phase transition in graph colouring, for a fixed number of colours, the lack of structure in random constraints does make a significant difference to asymptotic behaviour. In random binary CSPs, unless the number of values increases with the number of variables, there cannot asymptotically be a region where almost all problems are satisfiable, whereas in graph colouring problems, the number of values is the number of colours and by definition fixed.

The probability that a random instance can be solved without backtracking by a simple search algorithm has been used to give a lower bound on the crossover point, and this is novel in studies of phase transitions in CSPs. Although not a very tight bound and unlikely to be useful for predicting the exact location of the crossover point, it is adequate to show that there is a region of trivial satisfiability. Achlioptas [1] and Franco [4], in this issue, also derive lower bounds on satisfiability thresholds for k-SAT problems from the probability that a specified algorithm can show that an instance is satisfiable without backtracking.

Xu and Li [19] have considered a similar scheme for increasing the parameters of Model B with the number of variables. This is somewhat simpler than the scheme considered in this paper, although it has two parameters (γ, a) rather than one (a). [19] does not give any motivation for this scheme, but it has the great advantage that, except over a certain range of valuers of the parameters, the location of the asymptotic phase transition can be exactly determined. In Section 8, it is shown that in Model D, this scheme gives an asymptotic crossover point between fixed values of the constraint tightness even outside the range of parameters identified by Xu and Li. Further work to show whether Models B and D are asymptotically similar would be useful, as well as further study of the asymptotic phase transition where Xu and Li's results does not apply. It may be that the range of parameter values for which the asymptotic phase transition can be exactly located indicates qualitatively different asymptotic behaviour from the range where their result does not hold.

Acknowledgements

The author is a member of the APES research group (www.dcs.st-and.ac.uk/ ~apes) and wishes to thank the other members. Thanks are also due to Alan Slomson for the lemma given in Appendix A.3 and to Martin Dyer and Alan Frieze for the results on the width of random graphs.

Appendix A

A.1. Limit of $\hat{p}_2$ as $n \to \infty$

If we write $\hat{p}_2 = 1 - \hat{q}_2$, we have: $\hat{q}_2 = 0.5^{2/nd} m^{-2/d} \to m^{-2/d}$ as $m, d \to \infty$ with n. Hence, $\log \hat{q}_2 \to -2 \log m/d \to -\infty/\infty$ as $m, d \to \infty$ with n.

We can write m and d as functions of n. By l'Hôpital's rule, to find the limit of $\log m(n)/d(n)$ as $n \to \infty$, we find the limit of the ratio of the derivatives of $\log m(n)$ and $d(n)$.

m is given by

$$(1 - (1 - (1 - \tilde{\pi}_2)^d)^m)^{n-1} = 0.5.$$

Hence,

$$m(n) = \frac{\log(1 - 0.5^{1/(n-1)})}{\log(1 - (1 - \tilde{\pi}_2))^{d(n)})}.$$

Let $f(n) = \log m(n)$ and write $\tilde{\rho}_2 = 1 - \tilde{\pi}_2$. Then,

$$\lim_{n\to\infty} \frac{\log m(n)}{d(n)} = \lim_{n\to\infty} \frac{f(n)}{d(n)} = \lim_{n\to\infty} \frac{f'(n)}{d'(n)},$$

$$f'(n) = \frac{\tilde{\rho}_2^{d(n)} d'(n) \log \tilde{\rho}_2}{(1 - \tilde{\rho}_2^{d(n)}) \log(1 - \tilde{\rho}_2^{d(n)})} - \frac{0.5^{1/(n-1)} \log 2}{(n-1)^2 (1 - 0.5^{1/(n-1)}) \log(1 - 0.5^{1/(n-1)})}. \tag{A.1}$$

The denominator of the second term can be written as $(1 - 0.5^y)\log(1 - 0.5^y)/y^2$, where $y = 1/(n-1)$. This expression $\to 0/0$ as $n \to \infty$ and $y \to 0$, since $\lim_{x\to 0} x \log x = 0$.

Applying l'Hôpital's rule again, let $f_1(y) = (1 - 0.5^y)\log(1 - 0.5^y)$ and $g_1(y) = y^2$.

$$\begin{aligned} f_1'(y) &= -(1 + \log(1 - 0.5^y))0.5^y \log 0.5 \\ &\to -\infty \quad \text{as } y \to 0, \\ g_1'(y) &= 2y \\ &\to 0 \quad \text{as } y \to 0. \end{aligned}$$

Hence the second term in (A.1) $\to 0$ as $n \to \infty$, and we only need consider the first term, i.e. as $n \to \infty$,

$$\lim_{n\to\infty} f'(n) = \lim_{n\to\infty} \frac{\tilde{\rho}_2^{d(n)} d'(n) \log \tilde{\rho}_2}{(1 - \tilde{\rho}_2^{d(n)}) \log(1 - \tilde{\rho}_2^{d(n)})}$$

and

$$\lim_{n\to\infty} \frac{f'(n)}{d'(n)} = \frac{\tilde{\rho}_2^{d(n)} \log \tilde{\rho}_2}{(1 - \tilde{\rho}_2^{d(n)}) \log(1 - \tilde{\rho}_2^{d(n)})}$$

$$= -\log \tilde{\rho}_2 \quad \text{since} \quad \lim_{n\to\infty} \tilde{\rho}_2^{d(n)} = 0 \quad \text{and} \quad \lim_{x\to 0} \frac{x}{\log(1-x)} = -1.$$

So,

$$\lim_{n\to\infty} \log \hat{q}_2 = \lim_{n\to\infty} \frac{-2 \log m(n)}{d(n)} = 2 \log \tilde{\rho}_2$$

and

$$\lim_{n\to\infty} \hat{p}_2 = 1 - \tilde{\rho}_2^2 = 1 - (1 - \tilde{\pi}_2)^2.$$

A.2. Limit of m/n

Using the notation of Appendix A.1, since $d = a \log n + c$:

$$\tilde{\rho}_2^d = n^{a \log \tilde{\rho}_2} \tilde{\rho}_2^c.$$

We can write

$$\frac{m}{n} = \frac{-\log(1 - 0.5^{1/(n-1)})}{-n \log(1 - n^{a \log \tilde{\rho}_2} \tilde{\rho}_2^c)}.$$

As $n \to \infty$, the numerator of this $\to \infty$ and the denominator is indeterminate.

Applying l'Hôpital's rule to the denominator, let $f_2(n) = -\log(1 - n^{a \log \tilde{\rho}_2} \tilde{\rho}_2^c)$ and $g_2(n) = 1/n$.

$$\frac{f_2'(n)}{g_2'(n)} = \frac{-\tilde{\rho}_2^c a \log \tilde{\rho}_2 n^{a \log \tilde{\rho}_2 + 1}}{1 - n^{a \log \tilde{\rho}_2} \tilde{\rho}_2^c}.$$

Since $a > 0$ and $0 < \tilde{\rho}_2 < 1$, $a \log \tilde{\rho}_2 < 0$.

Hence,

$$\lim_{n\to\infty} \frac{f_2'(n)}{g_2'(n)} = \infty \quad \text{if } a \log \tilde{\rho}_2 + 1 > 0$$

$$= 0 \quad \text{if } a \log \tilde{\rho}_2 + 1 < 0$$

$$= -\tilde{\rho}_2^c a \log \tilde{\rho}_2 \quad \text{if } a \log \tilde{\rho}_2 + 1 = 0$$

Hence, if $a \log \tilde{\rho}_2 + 1 \leqslant 0$, $m/n \to \infty$ as $n \to \infty$. It still remains to show what the limit is if $a \log \tilde{\rho}_2 + 1 > 0$.

Let $f_3(n)=\log(1-0.5^{1/(n-1)})$ and $g_3(n)=n\log(1-n^{a\log\tilde{\rho}_2}\tilde{\rho}_2^c)$. Then,

$$f_3'(n)=\frac{0.5^{1/(n-1)}\log 0.5}{(1-0.5^{1/(n-1)})(n-1)^2}$$

$$\to 0 \quad \text{as } n\to\infty \text{ since } \lim_{x\to\infty} x(1-0.5^{1/x})=\log 2,$$

$$g_3'(n)=\log(1-n^{a\log\tilde{\rho}_2}\tilde{\rho}_2^c)-\frac{n^{a\log\tilde{\rho}_2}\tilde{\rho}_2^c a\log\tilde{\rho}_2}{1-n^{a\log\tilde{\rho}_2}\tilde{\rho}_2^c}$$

$$\to 0 \quad \text{as } n\to\infty \text{ since } a\log\tilde{\rho}_2<0,$$

$$f_3''(n)=\frac{0.5^{1/(n-1)}\log 2}{(n-1)^3(1-0.5^{1/(n-1)})}\left(2-\frac{\log 2}{(n-1)(1-0.5^{1/(n-1)})}\right)$$

$$\to \frac{\log 2}{(n-1)^2} \quad \text{as } n\to\infty,$$

$$g_3''(n)=-\frac{\tilde{\rho}_2^c a\log\tilde{\rho}_2 n^{a\log\tilde{\rho}_2-1}(1+a\log\tilde{\rho}_2-n^{a\log\tilde{\rho}_2}\tilde{\rho}_2^c)}{(1-n^{a\log\tilde{\rho}_2}\tilde{\rho}_2^c)^2}$$

$$\to -\tilde{\rho}_2^c a\log\tilde{\rho}_2(1+a\log\tilde{\rho}_2)n^{a\log\tilde{\rho}_2-1},$$

$$\frac{f_3''(n)}{g_3''(n)}\to -\frac{\log 2}{\tilde{\rho}_2^c a\log\tilde{\rho}_2(1+a\log\tilde{\rho}_2)(n-1)^2 n^{a\log\tilde{\rho}_2-1}}$$

and $(n-1)^2 n^{a\log\tilde{\rho}_2-1}\to n^{a\log\tilde{\rho}_2+1}\to\infty$ since $a\log\tilde{\rho}_2+1>0$ and $a\log\tilde{\rho}_2<0$. Hence,

$$\lim_{n\to\infty}\frac{f_3''(n)}{g_3''(n)}=0 \quad \text{if } a\log\tilde{\rho}_2+1>0.$$

Combining these results,

$$\lim_{n\to\infty}\frac{m}{n}=0 \quad \text{if } a\log\tilde{\rho}_2+1>0$$

$$=\infty \quad \text{if } a\log\tilde{\rho}_2+1\leqslant 0.$$

A.3. Limit of p_{sat} in Xu and Li's scheme

From the discussion above,

$$p_{\text{sat}}\geqslant \Pr(\text{backtrack-free search})\geqslant (1-(1-q_2^d)^m)^{n-1}.$$

Substituting $m=n^\gamma$, $d=a\log n$:

$$p_{\text{sat}}\geqslant (1-(1-q_2^{a\log n})^{n^\gamma})^{n-1}$$

$$=(1-(1-n^{a\log q_2})^{n^\gamma})^{n-1}.$$

Suppose q_2 is fixed, and $n \to \infty$. First consider $(1 - n^{a \log q_2})^{n^\gamma}$.

$$\lim_{n\to\infty} n^\gamma = \infty,$$

$$\lim_{n\to\infty} (1 - n^{a \log q_2}) = 1 \quad \text{since } a \log q_2 < 0.$$

Let $f_4(n) = \log(1 - n^{a \log q_2})$ and $g_4(n) = n^{-\gamma}$.

$$f_4'(n) = -\frac{a \log q_2 n^{a \log q_2 - 1}}{1 - n^{a \log q_2}},$$

$$g_4'(n) = -\gamma n^{-\gamma-1},$$

$$\frac{f_4'(n)}{g_4'(n)} = \frac{a \log q_2 n^{a \log q_2 + \gamma}}{\gamma(1 - n^{a \log q_2})},$$

$$\begin{aligned}\lim_{n\to\infty} n^\gamma \log(1 - n^{a \log q_2}) = \lim_{n\to\infty} \frac{f_4'(n)}{g_4'(n)} &= -\infty \quad \text{if } a \log q_2 + \gamma > 0 \\ &= -1 \quad \text{if } a \log q_2 + \gamma = 0 \\ &= 0 \quad \text{if } a \log q_2 + \gamma < 0.\end{aligned}$$

Hence,

$$\begin{aligned}\lim_{n\to\infty} (1 - n^{a \log q_2})^{n^\gamma} &= 0 \quad \text{if } a \log q_2 + \gamma > 0 \\ &= \mathrm{e}^{-1} \quad \text{if } a \log q_2 + \gamma = 0 \\ &= 1 \quad \text{if } a \log q_2 + \gamma < 0.\end{aligned}$$

When $a \log q_2 + \gamma > 0$, since $\lim_{n\to\infty} (1 - n^{a \log q_2})^{n^\gamma} = 0$,

$$\lim_{n\to\infty} (1 - (1 - n^{a \log q_2})^{n^\gamma})^{n-1} = \lim_{n\to\infty} (1 - (1 - n^{a \log q_2})^{n^\gamma})^n.$$

Write $c = a \log q_2$ and let $f_5(n) = \log(1 - (1 - n^c)^{n^\gamma})$ and $g_5(n) = 1/n$. Then $\lim_{n\to\infty} f_5(n) = 0$ and $\lim_{n\to\infty} g_5(n) = 0$.

$$f_5'(n) = \frac{n^{\gamma-1}(1 - n^c)^{n^\gamma} \{cn^c/(1 - n^c) - \gamma \log(1 - cn^c)\}}{1 - (1 - n^c)^{n^\gamma}},$$

$$g_5'(n) = -1/n^2,$$

$$\frac{f_5'(n)}{g_5'(n)} = -\frac{n^{\gamma+1}(1 - n^c)^{n^\gamma} \{cn^c/(1 - n^c) - \gamma \log(1 - cn^c)\}}{1 - (1 - n^c)^{n^\gamma}}.$$

We already know that $\lim_{n\to\infty}(1-n^c)^{n^\gamma}=0$ since $c=a\log q_2<0$ and $c+\gamma>0$. By the lemma below,

$$\lim_{n\to\infty} n^{\gamma+1}(1-n^c)^{n^\gamma}=0.$$

Hence,

$$\lim_{n\to\infty}\frac{f_5'(n)}{g_5'(n)}=0$$

and therefore

$$\lim_{n\to\infty}(1-(1-n^{a\log q_2})^{n^\gamma})^{n-1}=1 \quad \text{for } a\log q_2+\gamma>0$$

and

$$\lim_{n\to\infty} p_{\text{sat}}=1 \quad \text{if } a\log q_2+\gamma>0, \text{ i.e. if } p_2<1-\mathrm{e}^{-\gamma/a}.$$

Lemma A.1. *For $a,b,c>0$ and $a>b$,*

$$\lim_{x\to\infty} x^c(1-x^{-b})^{x^a}=0.$$

Proof. For $0<h<1$, $\log(1-h)=-h-(h^2/2)-(h^3/3)-\cdots$ Hence $1/h\log(1-h)=-1-(h/2)-(h^2/3)-\cdots<-1$ and so $(1-h)^{1/h}<\mathrm{e}^{-1}$.

For $x>1$ and $b>0$, $0<x^{-b}<1$, and hence $(1-x^{-b})^{x^b}<\mathrm{e}^{-1}$ and therefore:

$$(1-x^{-b})^{x^a}=((1-x^{-b})^{x^b})^{x^{a-b}}<\mathrm{e}^{-x^{a-b}}. \tag{A.2}$$

For all $k\geqslant 1$, $\lim_{x\to\infty}x^k\mathrm{e}^{-x}=0$. Since $a-b>0$,

$$\lim_{x\to\infty}(x^{a-b})^k\mathrm{e}^{-x^{a-b}}=0 \quad \text{for all } k\geqslant 1. \tag{A.3}$$

Also, as $a-b>0$, there exists k_0 such that $c<(a-b)k_0$, and hence:

$$\begin{aligned} x^c(1-x^{-b})^{x^a} &< x^{(a-b)k_0}(1-x^{-b})^{x^a} \\ &< x^{(a-b)k_0}\mathrm{e}^{-x^{a-b}} \quad \text{by (A.2)} \end{aligned}$$

Hence, by (A.3)

$$\lim_{x\to\infty} x^c(1-x^{-b})^{x^a}=0.$$

References

[1] D. Achlioptas, Lower bounds for random 3-SAT via differential equations, Theoret. Comput. Sci. 265 (this vol.) (2001) 159–185.

[2] D. Achlioptas, L.M. Kirousis, E. Kranakis, D. Krizanc, M.S.O. Molloy, Y.C. Stamatiou, Random constraint satisfaction—a more accurate picture, in: G. Smolka (Ed.), Principles and Practice of Constraint Programming—CP97, Lecture Notes in Computer Science, Vol. 1330, Springer, Berlin, 1997, pp. 107–120.

[3] P. Cheeseman, B. Kanefsky, W. Taylor, Where the really hard problems are, in: Proc. IJCAI-91, Vol. 1, 1991, pp. 331–337.
[4] J. Franco, Results related to threshold phenomena research in satisfiability: lower bounds, Theoret. Comput. Sci. 265 (this vol.) (2001) 147–157.
[5] E.C. Freuder, A sufficient condition for backtrack-free search, J. ACM 29 (1982) 24–32.
[6] D. Frost, R. Dechter, In search of the best constraint satisfaction search, in: Proc. AAAI'94, 1994, pp. 301–306.
[7] I. Gent, E. MacIntyre, P. Prosser, B. Smith, T. Walsh, Random constraint satisfaction: flaws and structure, Research Report 98.23, School of Computer Studies, University of Leeds, 1998. (Revised following reviewers' comments for Constraints).
[8] C.P. Gomes, B. Selman, N. Crato, Heavy-tailed distributions in combinatorial search, in: G. Smolka (Ed.), Principles and Practice of Constraint Programming—CP97, Lecture Notes in Computer Science, Vol. 1330, Springer, Berlin, 1997, pp. 121–135.
[9] S.A. Grant, B.M. Smith, The phase transition behaviour of maintaining arc consistency, Research Report 95.25, School of Computer Studies, University of Leeds, September, 1995.
[10] S.A. Grant, B.M. Smith, The phase transition behaviour of maintaining arc consistency, in: W. Wahlster (Ed.), Proc. ECAI'96, 1996, pp. 175–179.
[11] R. Haralick, G. Elliott, Increasing tree search efficiency for constraint satisfaction problems, Artificial Intelligence 14 (1980) 263–313.
[12] T. Hogg, C.P. Williams, The hardest constraint problems: a double phase transition, Artificial Intelligence 69 (1994) 359–377.
[13] E. MacIntyre, P. Prosser, B. Smith, T. Walsh, Random constraint satisfaction: theory meets practice, in: M. Maher, J.-F. Puget (Ed.), Principles and Practice of Constraint Programming—CP98, Lecture Notes in Computer Science, Vol. 1520, Springer, Berlin, 1998, pp. 325–339.
[14] B. Pittel, J. Spencer, N. Wormald, Sudden emergence of a giant k-core in a random graph, J. Combin. Theory (B) 67 (1996) 111–151.
[15] P. Prosser, An empirical study of phase transitions in binary constraint satisfaction problems, Artificial Intelligence 81 (1996) 81–109.
[16] B.M. Smith, M.E. Dyer, Locating the phase transition in constraint satisfaction problems, Artificial Intelligence 81 (1996) 155–181. (Special Issue on Frontiers in Problem Solving: Phase Transitions and Complexity.)
[17] B.M. Smith, S.A. Grant, Modelling exceptionally hard constraint satisfaction problems., in: G. Smolka (Ed.), Principles and Practice of Constraint Programming—CP97, Lecture Notes in Computer Science, Vol. 1330, Springer, Berlin, 1997, pp. 182–195.
[18] C. Williams, T. Hogg, Exploiting the deep structure of constraint problems, Artificial Intelligence 70 (1994) 73–117.
[19] K. Xu, W. Li, Exact phase transitions in random constraint satisfaction problems, J. AI Res. 12 (2000) 93–103.