Random Constraint Satisfaction: Flaws and Structure*

Ian P. Gent[1], Ewan MacIntyre[2], Patrick Prosser[3], Barbara M. Smith[4], and Toby Walsh[5]

[1] School of Computer Science, University of St Andrews, St Andrews, Fife KY16 9SS, United Kingdom. Email ipg@dcs.st-and.ac.uk

[2] Cisco Systems (Scotland) Ltd., Broadwood Business Park, Cumbernauld, Glasgow G68 9LE, United Kingdom. Email emacinty@cisco.com

[3] Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, United Kingdom. Email pat@dcs.gla.ac.uk

[4] School of Computing & Mathematics, University of Huddersfield, Huddersfield, HD1 3DH, United Kingdom. Email b.m.smith@hud.ac.uk

[5] Department of Computer Science, University of York, York YO10 5DD, United Kingdom. Email tw@cs.york.ac.uk

Abstract. A recent theoretical result by Achlioptas *et al.* shows that many models of random binary constraint satisfaction problems become trivially insoluble as problem size increases. This insolubility is partly due to the presence of 'flawed variables', variables whose values are all 'flawed' (or unsupported). In this paper, we analyse how seriously existing work has been affected. We survey the literature to identify experimental studies that use models and parameters that may have been affected by flaws. We then estimate theoretically and measure experimentally the size at which flawed variables can be expected to occur. To eliminate flawed values and variables in the models currently used, we introduce a 'flawless' generator which puts a limited amount of structure into the conflict matrix. We prove that such flawless problems are not trivially insoluble for constraint tightnesses up to 1/2. We also prove that the standard models B and C do not suffer from flaws when the constraint tightness is less than the reciprocal of domain size. We consider introducing types of structure into the constraint graph which are rare in random graphs and present experimental results with such structured graphs.

Keywords: Constraint satisfaction, random problems, benchmarking, phase transitions

* This work was supported by EPSRC awards GR/L/24014 and GR/K/65706 and was completed while the fourth author was employed at the University of Leeds. The authors are all members of the APES research group, http://www.dcs.st-and.ac.uk/~apes, and we wish to thank other members of the group. We especially thank Yeo Shao Hong for identifying an error in an earlier version of this paper, and Joseph Culberson for extensive help correcting the error. We also thank reviewers of this journal for their helpful comments.

Submission to Constraints

Corresponding author: Ian Gent, School of Computer Science, University of St Andrews, St Andrews, Fife KY16 9SS, United Kingdom. Email ipg@dcs.st-and.ac.uk, URL http://www.dcs.st-and.ac.uk/~ipg, Tel +44 1334 463247, Fax +44 1334 463278

1 Introduction

One of the most exciting areas in AI in recent years has been the study of phase transition behaviour. In a seminal paper that inspired many later researchers, Cheeseman, Kanefsky, and Taylor showed empirically that the hardest instances to solve in a number of NP-complete problems often occur around a rapid transition in solubility [7]. Problems from such transitions in solubility are routinely used to benchmark algorithms for many different NP-complete problems. Experimental results about phase transition behaviour have come thick and fast since the publication of [7]. For example, in random 3-SAT, the phase transition was quickly shown to occur when the ratio of clauses to variables is approximately 4.3 [45]. Unfortunately, theory has often proved more difficult. A recent result proves that the width of the phase transition in random 3-SAT narrows as problems increase in size [15]. However, we only have rather loose but hard won bounds on its actual location [16, 37]. For random constraint satisfaction problems (CSPs), Achlioptas *et al.* recently provided a more negative theoretical result [1]. They show that, as the number of variables increases, the conventional random models produce problems which almost surely contain flawed variables and are therefore trivially insoluble. Thus, these models do not have an asymptotic phase transition over most of their parameter space. This paper studies the impact of this theoretical result on experimental studies. We show how to add structure to random problems to overcome such flaws, as well as to make them more representative of problems met in practice.

The paper can be broadly divided into two parts reflecting our subtitle: flaws and structure. In the first part of the paper, from Section 4 to Section 6, we analyse the consequences of Achlioptas *et al.*'s discovery that the most commonly used methods of generating random problems suffer from flaws. In Section 4 we survey the literature, showing that many past studies may indeed have been compromised by flaws. In Section 5 we estimate theoretically the likelihood of flaws arising and in Section 6 confirm these estimates theoretically. In the second part of the paper, from Section 7 to Section 10 we show how structure can be added to generation methods to eliminate flaws and we consider using constraint graphs with specific structure, to make random problems more representative of problems met in practice. In Section 7 we introduce a new flawless method of generating binary CSPs, and show how it can be used with existing models. In Section 8, we justify the name 'flawless' by proving that, asymptotically, flawless instances are not trivially insoluble. A corollary of this result is that problems from the standard models B and C are not trivially insoluble in the limit, if the constraint tightness $< 1/m$, where m is the domain size. We also report empirical results using both the flawless method (Section 9) and constraint graphs which have specific structure, rather than being purely random (Section 10).

2 Constraint Satisfaction

A binary constraint satisfaction problem consists of a set of variables, each with a domain of values, and a set of binary constraints. Each constraint defines the incompatible values for a pair of variables. Each assignment of values to variables ruled out is called a nogood. We can describe the constraint between the variables v_x and v_y by a conflict matrix. This is a 0-1 matrix with 0 in the (i, j) entry iff the ith value for v_x is incompatible with the jth value for variable v_y and 1 otherwise. Associated with each problem is a constraint graph. This has variables as vertices and edges between variables that appear together in a constraint. The constraint satisfaction decision problem is to decide if there is an assignment of values to variables such that none of the constraints are violated.

Randomly-generated binary CSPs have been widely used experimentally, for instance to compare different solution algorithms. Most experimental and theoretical studies use one of four simple models of random problems. In each of these models, we generate a constraint graph G, and then for each edge in this graph, we choose pairs of incompatible values for the associated conflict matrix. The models differ in how we generate the constraint graph and how we choose incompatible values. In each case, we can describe problems 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.

model A: we independently select each one of the $n(n-1)/2$ possible edges in G with probability p_1, and for each selected edge we pick each one of the m^2 possible pairs of values, independently with probability p_2, as being incompatible;

model B: we randomly select exactly $p_1 n(n-1)/2$ edges for G, and for each selected edge we randomly pick exactly $p_2 m^2$ pairs of values as incompatible;

model C: we select each one of the $n(n-1)/2$ possible edges in G independently with probability p_1, and for each selected edge we randomly pick exactly $p_2 m^2$ pairs of values as incompatible;

model D: we randomly select exactly $p_1 n(n-1)/2$ edges for G, and for each selected edge we pick each one of the m^2 possible pairs of values, independently with probability p_2, as being incompatible;

While we use the same notation p_1 and p_2 in each model, note that in some cases the value is used as a proportion, and in others as a probability. For example in model D, p_1 is used as a proportion but p_2 is used as a probability.

3 The Problem with Random Problems

Achlioptas *et al.* [1] identify a shortcoming of all four random models. They prove that if $p_2 \geq 1/m$ then, as $n \to \infty$, there almost surely exists a *flawed* variable, one for which every value is *flawed*. A value for a variable is flawed if, when the value is assigned to the variable, there exists an adjacent variable in the constraint graph that cannot be assigned a value without violating the constraint between the two variables. A value for a variable is *supported* if it is not flawed. A problem with a flawed variable cannot have a solution. They argue that therefore "the currently used models are asymptotically uninteresting except, perhaps, for a small region of their parameter space" (when $p_2 < 1/m$). Further, they claim that "the threshold-like picture given by experimental results [with these models] is misleading, since the problems with defining parameters in what is currently perceived as the underconstrained region (because a solution can be found fast) are in fact overconstrained for large n (obviously, larger than the values used in experiments)". As they point out, this result does not apply to problems in which the constraints are not completely random but have a certain amount of structure. For example, if conflict matrices only have 0's on the diagonal then neighbouring variables in the constraint graph must take different values. These are graph colouring problems, which have good asymptotic properties.

Achlioptas *et al.* [1] propose an alternative random problem class, model E, which they show has better asymptotic properties than models A to D. This model does not separate the generation of the constraint graph from the selection of the nogoods.

model E: we select uniformly, independently and with repetitions, $pm^2n(n-1)/2$ nogoods out of the $m^2n(n-1)/2$ possible.

They show that there is a range of parameter values for which instances generated by this model almost surely have a solution, and a range of parameters for which instances almost surely do not have a solution, and hence this model does not suffer from the deficiencies of the other models discussed earlier.

In passing, we note that model E is not entirely novel since Williams and Hogg study random problems both with a fixed number of nogoods picked uniformly from the set of all possible nogoods, and with a uniform probability of including a nogood [58]. As Achlioptas *et al.* themselves observe [1], the expected number of repetitions in model E is usually insignificant (for instance, it is $O(1)$ when the number of nogoods is $\Theta(n)$), and repetitions are only allowed in order to simplify the theoretical analysis. The differences between model E and the models of Williams and Hogg are therefore likely to be slight.

More recently, Xu and Li [59] and Smith [56] have shown that variants of models B and D respectively can exhibit interesting asymptotic behaviour. In these variants, both the number of values, m, and the number of constraints increase with n in a specified way, dependent on additional parameters. Further, Xu and Li show that the location of the asymptotic phase transition can be determined exactly for a certain range of these parameters.

Model E was proposed in order to deal with the difficulty that, as the number of variables increases, asymptotically these models produce trivially insoluble problems. It might therefore be natural to wonder whether model E should be used as an experimental problem generator, in preference to the standard models.

However, models A to D generate the constraint graph and constraint matrices separately, whereas in model E the constraint graph emerges from the nogoods selected, and cannot be independently controlled. The standard models therefore give much greater flexibility in the range of instance types that can be generated. A particular shortcoming of model E as a source of benchmark problems is that it generates complete constraint graphs for quite small values of p, even though each constraint contains just a few nogoods. It is hard therefore to generate sparse constraint graphs with tight constraints. In model E, we randomly select $pm^2n(n-1)/2$ nogoods independently and with repetitions. By a coupon collector's argument, we expect a complete constraint graph when $p \approx \log(n(n-1)/2)/m^2$. For example, for $n = 20$, $m = 10$, we expect a complete constraint graph when $p \approx 0.052$. With a larger number of nogoods, there is a very small probability that the constraint graph is not complete. Hence, although model E has good asymptotic properties, it is not suitable for use as a problem generator at the small problem sizes which are feasible for experimental studies.

4 Past Experimental Practice

Achlioptas *et al.*'s result, that models A to D are "asymptotically uninteresting", does not apply to random problems from models B and C for which $p_2 < 1/m$. Indeed, as we prove in Section 8, such problems are not trivially insoluble in the limit. To study the practical significance of this restriction, we surveyed the literature from 1994 (when phase transition experiments with random constraint satisfaction problems first started to appear) to 1997, covering all papers in the proceedings of the CP, AAAI, ECAI and IJCAI conferences which gave details of experiments on random constraint satisfaction problems. The results of this survey are summarized in Table 1. Just over a quarter of the papers include some set of problems to which the results of [1] do not apply. Most commonly, these exceptions are random problems with $m = 3$ and $p_2 = 1/9$ or $2/9$, using Model B. However, all of the papers which included inapplicable problem sets also used some sets with $p_2 \geq 1/m$. In conclusion, all published experiments which we have considered use ensembles of problems that satisfy the preconditions of Achlioptas *et al.*'s result.

5 Probability of Flawed Variables

As Achlioptas *et al.* themselves suggest [1], most previous experimental studies will not have been greatly influenced by the existence of flawed variables since the number of variables is usually too small. Using the Markov inequality, they give a first moment bound on the probability of a flawed variable,

$$\Pr\{\text{problem has a flawed variable}\} \leq n(1-(1-p_2^m)^n)^m$$

For example, for the $\langle n, 10, 1, 1/2\rangle$ problem class, they calculate that the probability of a flawed variable is less than 10^{-5} even for n as large as 200. At what size of problem and sample do flawed variables start to occur?

By making a few simplifying assumptions, we can estimate the probability of a flawed variable with reasonable accuracy. Our first assumption is that each variable is connected to exactly $p_1(n-1)$ others. In practice, some variables have a greater degree, whilst others have a lesser degree. Fortunately, our experiments

Conference	Author initials	Reference	Model	$\langle n, m\rangle$	B/C & $p_2 < 1/m$?
AAAI-94	[DF,RD]	[17]	B	$\langle 25-250, 3\rangle$	$p_2 = 1/9, 2/9$
	[DF,RD]	[18]	B	$\langle 25-275, 3\rangle$	$p_2 = 1/9, 2/9$
				$\langle 15-60, 6\rangle$	$p_2 = 4/36$
				$\langle 15-35, 9\rangle$	no
	[NY,YO,HH]	[60]	B	$\langle 20, 10\rangle$	no
ECAI-94	[PP]	[47]	D	$\langle 20, 10\rangle$, $\langle 20, 20\rangle$, $\langle 30, 10\rangle$	no
	[BMS]	[51]	B	$\langle 8, 10\rangle$	no
	[DL]	[42]	B	$\langle 10, 20\rangle$	no
	[DS,ECF]	[49]	A	$\langle 50, 8\rangle$	no
CP-95	[IPG,EM,PP,TW]	[26]	B	$\langle 10-110, 3\rangle$	$p_2 = 2/9$
				$\langle 10, 10\rangle$, $\langle 20, 10\rangle$, $\langle 10, 5-50\rangle$, ...	no
	[JL,PM]	[39]	A	$\langle 10, 10\rangle$	no
	[FB,PR]	[3]	B	$\langle 25, 3\rangle$	$p_2 = 1/9$
				$\langle 35, 6\rangle$, $\langle 50, 6\rangle$	$p_2 = 4/36$
				$\langle 15, 9\rangle$, $\langle 35, 9\rangle$	no
	[FB,AG]	[2]	B	$\langle 25, 3\rangle$	$p_2 = 2/9$
				$\langle 25, 6\rangle$, $\langle 15, 9\rangle$	no
IJCAI-95	[ECF,PDH]	[14]	A	$\langle 50, 8\rangle$	no
	[DF,RD]	[19]	B	$\langle 125, 3\rangle$	$p_2 = 1/9$
				$\langle 35, 6\rangle$	$p_2 = 4/36$
				$\langle 250, 3\rangle$, $\langle 50, 6\rangle$, $\langle 35, 9\rangle$, ...	no
	[PM,JL]	[44]	D	$\langle 10, 10\rangle$, $\langle 20, 10\rangle$, $\langle 30, 10\rangle$	no
	[KK,RD]	[35]	B	$\langle 100, 8\rangle$	no
	[BMS,SAG]	[55]	B	$\langle 20, 10\rangle$, $\langle 50, 10\rangle$	no
AAAI-96	[AC,PJ]	[8]	B	$\langle 16, 8\rangle$, $\langle 32, 8\rangle$	no
	[ECF,CDE]	[13]	B	$\langle 100, 6\rangle$	no
	[IPG,EM,PP,TW]	[27]	B	$\langle 20, 10\rangle$	no
	[KK,RD]	[36]	B	$\langle 100, 8\rangle$, $\langle 125, 6\rangle$, $\langle 150, 4\rangle$	no
CP-96	[CB,JCR]	[4]	B	$\langle 35, 6\rangle$	$p_2 = 4/36$
				$\langle 125, 3\rangle$, $\langle 350, 3\rangle$	$p_2 = 1/9$
				$\langle 35, 9\rangle$, $\langle 50, 6\rangle$, $\langle 50, 20\rangle$, ...	no
	[DAC,JF,IPG,EM,NT,TW]	[9]	B	$\langle 20, 10\rangle$	no
	[IPG,EM,PP,BMS,TW]	[24]	B	$\langle 20-50, 10\rangle$	no
	[JL,PM]	[40]	B	$\langle 15, 5\rangle$	$p_2 = 1/25 - 4/25$
				$\langle 10, 10\rangle$	$p_2 = 1/100 - 9/100$
	[RJW]	[57]	A	$\langle 30, 5\rangle$, $\langle 100, 5\rangle$	no
ECAI-96	[JEB,EPKT,NRW]	[5]	B	$\langle 50, 10\rangle$	no
	[BC,GV,DM,PB]	[6]	B	$\langle 50, 10\rangle$, $\langle 20, 5\rangle$	no
	[SAG,BMS]	[34]	B	$\langle 30-70, 10\rangle$	no
	[ACMK,EPKT,JEB]	[38]	B	$\langle 30, 5\rangle$	$p_2 = 0.12$
				$\langle 40, 5\rangle$	$p_2 = 0.08$
				$\langle 60, 5\rangle$	$p_2 = 0.04$
				$\langle 10, 5\rangle$, $\langle 20, 5\rangle$, $\langle 10, 10\rangle$, ...	no
	[JL,PM]	[41]	B	$\langle 10, 10\rangle$	no
AAAI-97	[AM,SES,GS]	[43]	B	$\langle 6-12, 9\rangle$	no
	[DRG,WKJ,WSH]	[22]	B	$\langle 10, 5\rangle$	no
	[IPG,EM,PP,TW]	[28]	B	$\langle 10-120, 3\rangle$	$p_2 = 2/9$
				$\langle 10, 10-100\rangle$	no
	[DF,IR,LV]	[20]	B	$\langle 20, 4\rangle$	$p_2 = 0.125$
				$\langle 150, 3\rangle$	$p_2 = 0.222$
				$\langle 20-75, 6\rangle$, $\langle 20, 10\rangle$	no
CP-97	[IPG,JLU]	[29]	D	$\langle 10, 10\rangle$	no
	[IR,DF]	[48]	B	$\langle 100, 8\rangle$	no
	[DS,ECF]	[50]	B	$\langle 20, 20\rangle$, $\langle 40, 20\rangle$	no
	[BMS,SAG]	[54]	B	$\langle 10, 10\rangle$	no
	[PG,JKH]	[21]	B	$\langle 50, 10\rangle$, $\langle 100, 15\rangle$, $\langle 250, 25\rangle$, ...	no
	[RD,CB]	[10]	B	$\langle 100, 20\rangle$	no
	[IPG,EM, PP, PS, TW]	[23]	B	$\langle 20-70, 10\rangle$	no
IJCAI-97	[RD, CB]	[11]	B	$\langle 20, 10\rangle$	no

Table 1. Parameters and models used in some previous studies of random constraint satisfaction problems. The final column details studies in which model B or C was used and $p_2 < 1/m$. In the limit, such problem classes are not trivially insoluble.

show that this mean-field approximation does not introduce a large error into the estimate. We also assume independence between the probabilities that the different variables have at least one unflawed value. The probability that there are no flawed variables is then simply the product of the probabilities that the variables have at least one unflawed value. For model A problems, we have:

$$\begin{aligned}
&\Pr\{\text{problem has a flawed variable}\}\\
&= 1 - \Pr\{\text{there are no flawed variables}\}\\
&= 1 - (\Pr\{\text{a variable has at least one unflawed value}\})^n\\
&= 1 - (1 - \Pr\{\text{every value for the variable is flawed}\})^n\\
&= 1 - (1 - (\Pr\{\text{a value for the variable is flawed}\})^m)^n\\
&= 1 - (1 - (\Pr\{\text{value inconsistent with every value of an adjacent variable}\})^m)^n\\
&= 1 - (1 - (1 - \Pr\{\text{value consistent with a value of every adjacent variable}\})^m)^n\\
&= 1 - (1 - (1 - (\Pr\{\text{value consistent with a value of an adjacent variable}\})^{p_1(n-1)})^m)^n\\
&= 1 - (1 - (1 - (1 - \Pr\{\text{value inconsistent with every value of adjacent variable}\})^{p_1(n-1)})^m)^n\\
&= 1 - (1 - (1 - (1 - (\Pr\{\text{value inconsistent with a value of adjacent variable}\})^m)^{p_1(n-1)})^m)^n
\end{aligned}$$

For model A, the probability that a given value is inconsistent with a value of an adjacent variable is p_2. Hence, we obtain the estimate,

$$\Pr\{\text{problem has a flawed variable}\} = 1 - (1 - (1 - (1 - p_2^m)^{p_1(n-1)})^m)^n$$

A similar derivation can be made for model B problems, except that the last line can be omitted. Instead, we can calculate directly the value of the probability that there is a value inconsistent with every value of an adjacent variable. In model B each constraint matrix is picked uniformly from the $\binom{m^2}{p_2 m^2}$ possible matrices. If we assign a value to one of the variables involved in a constraint, then $\binom{m^2-m}{p_2 m^2 - m}$ of the possible constraints have nogoods that rule out all the values for the other variable. Hence, the probability that a particular value for a variable is inconsistent with every value for an adjacent variable is given by,

$$\Pr\{\text{value inconsistent with every value of adjacent variable}\} = \binom{m^2 - m}{p_2 m^2 - m} \Big/ \binom{m^2}{p_2 m^2}$$

Thus, for model B problems, the estimate for $\Pr\{\text{problem has a flawed variable}\}$ is:

$$1 - \left(1 - \left(1 - \left(1 - \binom{m^2 - m}{p_2 m^2 - m} \Big/ \binom{m^2}{p_2 m^2}\right)\right)^{p_1(n-1)}\right)^{m}\Bigg)^{n}$$

Note that we have assumed independence between the probabilities that the m different values for a given variable are flawed. The probability that every value for a variable is flawed is then simply the product of the probabilities that each individual value is flawed. Whilst this independence assumption is valid for model A, it is not strictly true for model B.

6 Occurrence of Flawed Variables

We can use these estimates for the probability of a flawed variable to determine when flawed variables will start to occur in experimental studies. To test the accuracy of the estimates and to compare them with the simpler first moment bound, we generated random problems using model B. We tested each instance for flawed variables. This test is linear in the size of the problem, so we were able to experiment with problems containing thousands of variables with large samples. Since flawed variables are more likely in dense constraint graphs, we generated problems with complete constraint graphs (i.e. with $p_1 = 1$). As in other studies (e.g. [33, 24]), we also generated a separate ensemble of problems in which the constraint graph has constant average degree, γ. That is, $p_1 = \gamma/(n-1)$. The constraint tightness for which the expected number of solutions is 1 is then constant as n increases; this constraint tightness is often a good predictor of the transition from soluble to insoluble problems ([52]). Empirically, the transition is observed to occur at roughly the same value of p_2 over a wide range of values of n. Keeping the average degree constant also reduces the probability of flawed variables occurring. In Table 2, we give the results for $\langle n, 10, 1, 1/2\rangle$ and $\langle n, 10, 19/(n-1), 1/2\rangle$ with n from 200 to 4000. In this (and indeed all the subsequent experiments) our estimate for the probability of a problem having a flawed variable is very close to the observed fraction of problems with flawed variables, and much closer than the first moment bound to the observed fraction of flawed variables.

With complete constraint graphs, flawed variables are observed in samples of 1000 when the problems have 500 or more variables. This is beyond the size of problems typically solved with systematic procedures but potentially within the reach of approximation or local search algorithms. By comparison, with constraint graphs of constant average degree, flawed variables are not observed in samples of 1000 even when the problems have thousands of variables. Because of the greater homogeneity of model B problems, we expect flawed variables to be less likely than in model A. Our estimates for the probability of a flawed variable support this conjecture. For example, for $\langle 1000, 10, 1, 1/2\rangle$ problems, our estimate for the probability that a model A problem has a flawed variable is 0.99986 whilst for a model B problem it is 0.275.

With constraint graphs of constant average degree, we can estimate the size of problems at which we expect to observe flawed variables. If $p_1 = \gamma/(n-1)$ and a fraction f of problems contain flawed variables then, by rearranging our estimates for the probability of a flawed variable, the number of variables n_f in model A problems is,

$$n_f = \frac{\log(1-f)}{\log(1-(1-(1-p_2^m)^\gamma)^m)}$$

and in model B problems,

$$n_f = \frac{\log(1-f)}{\log(1-(1-(1-\binom{m^2-m}{p_2m^2-m}/\binom{m^2}{p_2m^2})^\gamma)^m)}$$

n	sample size	fraction with flawed variables	estimate for Pr{flawed variable}	1st moment bound
200	10^6	0	$< 10^{-6}$	0.000006
500	10^4	0.0005	0.0006	0.0370
1000	10^3	0.272	0.275	> 1
1200	10^3	0.753	0.755	> 1
1500	10^3	1	0.999	> 1
2000	10^3	1	1.000	> 1
4000	10^3	1	1.000	> 1

(a) $\langle n, 10, 1, 1/2\rangle$

n	sample size	fraction with flawed variables	estimate for Pr{flawed variable}	1st moment bound
200	10^3	0	$< 10^{-3}$	0.000
500	10^3	0	$< 10^{-3}$	0.037
1000	10^3	0	$< 10^{-3}$	> 1
1500	10^3	0	$< 10^{-3}$	> 1
2000	10^3	0	$< 10^{-3}$	> 1
4000	10^3	0	$< 10^{-3}$	> 1

(b) $\langle n, 10, 19/(n-1), 1/2\rangle$

Table 2. The impact of flawed variables on model B problems with a domain size of 10 and: **(a)** complete constraint graphs; **(b)** constraint graphs of constant average degree.

For instance, for model B problems with similar parameters to those of Table 2 (i.e. $m = 10$, $\gamma = 19$ and $p_2 = 1/2$), $n_{1/1000} \approx 3.2 * 10^{17}$ and $n_{1/2} \approx 2.2 * 10^{19}$. That is, problems need more than 10^{17} variables before we expect to observe flawed variables in samples of 1000 problem instances, and more than 10^{19} variables before half can be expected to contain a flawed variable. As a consequence, at this domain size, constraint tightness, and degree of the constraint graph, experimental studies can safely ignore flawed variables.

With smaller domain sizes, we expect flawed variables to be more prevalent. To test this hypothesis, we generated problems with $m = 3$, $p_2 = 1/m$ and either complete constraint graphs or constraint graphs of constant average degree. Note that, for model B, $p_2 = 1/m$ is the smallest possible value which gives flawed variables. If $p_2 < 1/m$ then at least one value for each variable *must* be supported, as each constraint rules out strictly less than m possible values. Note also that problems with $m = 3$ and $p_2 = 1/m$ have the same domain size and constraint tightness as 3-colouring problems. Table 3 gives the results for $\langle n, 3, 1, 1/3\rangle$ and $\langle n, 3, 19/(n-1), 1/3\rangle$ with $n = 10$ to 2000. With complete constraint graphs, flawed variables occur with significant frequency in problems with as few as 20 variables. With constraint graphs of constant average degree, although flawed variables again occur in problems with as few as 20 variables, their frequency increases much more slowly with n. We need a thousand or more variables to ensure that problems almost always include a flawed variable. By comparison, with complete constraint graphs, we need just 60 or so variables.

Some of the experiments surveyed in Section 4 used random problems containing hundreds of variables with $m = 3$ and p_2 between 1/9 and 4/9. We performed a simple experiment to show that flawed values will have significantly influenced such experiments. We tested problems generated using model B with 100 variables, $p_2 = 4/9$, and 92 constraints, repeating one of the experiments reported by [18] at the 50% solubility point. In a sample of 100, *all* problems contained flawed values. On average a problem contained 26.7 flawed values (minimum 19, maximum 36). Four problems contained a flawed variable and were thus trivially insoluble. The extent to which these flaws influenced behaviour can also be seen in Section 9, where we compare the phase transition from model B with a new generation method which is guaranteed to give problems without flawed values or variables.

The papers in Table 1 were for the most part using experiments on random CSPs in order to compare the performance of different solution methods. The presence of flawed values or variables would favour methods which look for such flaws, and unless this is recognised by the experimenter can distort the conclusions.

n	sample size	fraction with flawed variables	estimate for Pr{flawed variable}	1st moment bound
10	10^3	0.006	0.011	0.311
20	10^3	0.143	0.156	> 1
30	10^3	0.504	0.536	> 1
40	10^3	0.869	0.882	> 1
50	10^3	0.987	0.990	> 1
60	10^3	1	1.000	> 1

(a) $\langle n, 3, 1, 1/3\rangle$

n	sample size	fraction with flawed variables	estimate for Pr{flawed variable}	1st moment bound
20	10^3	0.143	0.156	> 1
50	10^3	0.318	0.345	> 1
100	10^3	0.524	0.571	> 1
200	10^3	0.796	0.816	> 1
500	10^3	0.986	0.985	> 1
1000	10^3	0.999	1.000	> 1
2000	10^3	1	1.000	> 1

(b) $\langle n, 3, 19/(n-1), 1/3\rangle$

Table 3. The impact of flawed variables on model B problems with a small domain size and: **(a)** complete constraint graph; **(b)** constraint graph of constant average degree.

7 Flawless Random Problem Generation

Until this point in the paper, we have been analysing the effect of flawed values and variables on past experiments and existing models. Unfortunately no existing model meets the twin desiderata of allowing the flexibility of traditional models like A to D, with the good asymptotic property of model E. We will therefore introduce some simple variants of models A to D which have similar asymptotic properties to those holding for model E but allow us to generate the constraint graph and the conflict matrices independently as in models A to D. Aside from the absence of flaws, these models give problems very similar to those generated by the traditional models.

The reason that traditional models of random CSPs suffer asymptotically from trivial insolubility is that they allow *flawed* values. Flawed values can cause flawed variables, which in turn cause trivial insolubility. Since a flawed value is exactly a value without support across some constraint, simply insisting that all constraints are arc consistent guarantees that flawed values and variables cannot occur. It is easy to adapt all the traditional models by discarding and re-generating constraints which are not arc consistent. Unfortunately this does not give us an asymptotic guarantee against trivial insolubility, because simple cycles in a small part of the constraint graph might make a problem insoluble, and these may be sufficiently probable to lead asymptotically to trivial insolubility.

Instead we propose a new way of generating conflict matrices which we call 'flawless' since problems are guaranteed not to be trivially insoluble. The basic idea is that each value is supported by at least one unique value, i.e. at least one value which is not also required to support another value. We cannot then get a *chain reaction* in which support for values of several other variables disappears if we remove one value. We first introduce the flawless model and then prove its desirable asymptotic properties.

Definition 1 (Flawless). *A conflict matrix is* **flawless** *if there is a permutation* π *of* $1, 2, \ldots, m$ *such that all the pairs of values* $(1, \pi(1)), (2, \pi(2)), \ldots, (m, \pi(m))$ *are allowed.*

It is clear that a flawless matrix must be arc consistent, because the value $\pi(i)$ always supports value i. We mistakenly believed the converse, and are grateful to Yeo Shao Hong for suggesting the matrix in Figure 1 which contradicts this.[1] Insisting on flawless matrices gives us good asymptotic properties as we shall prove in the next section. First, however, we illustrate how easy it is to adapt any of the existing models A to D to generate flawless conflict matrices.

For models B and C, in which all conflicts for a constraint are selected together, there is a simple way to generate flawless instances. Given a pair of variables between which we wish to construct a constraint, we choose a random

[1] Unfortunately the mistake is present in the original Research Report version of this paper [25], so we ask readers to use definitions from this paper and not the original report. In particular, note that our definition of flawless here corresponds to the definition of "strongly flawless" in the original.

		v_2		
		1	2	3
	1	0	0	1
v_1	2	0	0	1
	3	1	1	0

Fig. 1. A conflict matrix which is arc consistent but not flawless. Since a permutation of $1, \ldots m$ corresponds to a placement of m non-attacking rooks on a m by m chess board, we can see that the problem is not flawless by trying to place 3 rooks on the squares where there are 1s in the matrix. In this case two rooks in the third row and in the third column leave no space for a third rook.

		v_2			
		1	2	3	4
	1	0	0	1	0
v_1	2	1	0	0	0
	3	0	0	0	1
	4	0	1	0	0

		v_2			
		1	2	3	4
	1	0	1	1	0
v_1	2	1	0	0	0
	3	1	1	1	1
	4	1	1	0	0

Fig. 2. The first conflict matrix shows a flawless constraint arising from the permutation $3, 1, 4, 2$. Even though the tightness is $3/4$, every value for both variables is supported. The second conflict matrix shows a flawless constraint with tightness $7/16$ derived from the first conflict matrix by choosing randomly 7 of the 12 conflicts.

permutation π of $1, 2, 3 \ldots, m$. The set of goods based on this permutation is simply $\{(1, \pi(1)), (2, \pi(2)), (3, \pi(3)), \ldots, (m, \pi(m))\}$. A conflict matrix that contains these goods cannot give a flawed value. We therefore remove the goods just chosen from the set of all possible conflicts and choose $p_2 m^2$ elements randomly from the remainder. An example is shown in Figure 2. For models A and D the process is similar, except that having removed a set of goods, we increase p_2 to $mp_2/(m-1)$ before selecting conflicts.

8 Theory of Flawlessness

We now prove some asymptotic results about flaws. We will show the unexpected result that conflict matrices generated by the flawless variants of models B and C are not trivially insoluble for all p_2 up to $p_2 < 1/2$. As a corollary, the standard models B and C do not suffer asymptotically from trivial insolubility whenever $p_2 < 1/m$. Whilst these results do not apply directly to the flawless variants of models A and D, they can be made to by adapting those models to reject candidate conflict matrices in which the proportion of conflicts selected $\geq 1/2$.

The proof of our main result proceeds in a manner similar to that of Achlioptas *et al.*'s proof of the analogous result for model E: we show that a constraint graph in which each component has at most one cycle is guaranteed to be soluble, and appeal to a graph theoretic result to show that such constraint graphs occur with a ratio of constraints to variables bounded away from zero.

Lemma 1. *If the constraint graph of a flawless binary CSP (without unary constraints) is a forest, the instance has a solution. Furthermore, for each variable in the instance, and each value in its domain, there is a solution in which it takes that value.*

Proof. A flawless binary CSP is necessarily strongly arc consistent, since we assume that there are no unary constraints. A forest has width 1, and we can apply Freuder's Theorem [12] to show that search is backtrack free. Because there are no unary constraints, we can choose the first variable and its value arbitrarily, and extend it to a solution.

Given flawless constraint matrices, we have the following theorem. The fact that this result extends to such a large value of p_2, i.e. $\frac{1}{2}$, is rather surprising.

Theorem 1. *If a binary CSP with uniform domain size contains only flawless constraints with $p_2 < 1/2$, and each component in the constraint graph contains at most one cycle, the instance is soluble.*

Proof. If the constraint graph is acyclic, Lemma 1 applies, and with the result for one component we can apply it to each component of a graph in turn. So for the rest of the proof we consider a constraint graph containing a single component which contains exactly one cycle. We will show that there is an assignment which satisfies all the constraints in the cycle. Having done that, the assignment can be extended to the entire component by giving each variable in the cycle the relevant value, removing all constraints in the cycle, and appealing to Lemma 1. Thus we have reduced the proof to showing that if a cycle of flawless constraints is insoluble, $p_2 \geq 1/2$.

For the general case of a cycle of length three or more, consider an insoluble cycle, say $v - x_0 - x_1 - \ldots - x_k - v$. (In a triangle, $k = 1$.) Consider the constraints $v - x_0$ and $x_k - v$. We claim that these two constraint matrices must, between them, contain at least m^2 conflicts, so $p_2 \geq \frac{1}{2}$. The proof will be completed by justifying the claim.

If there is no satisfying assignment, the value $v = 1$ in particular must be impossible. Some number r of conflicts involving $v = 1$ and the constraint $v - x_0$ rules out r values for x_0, leaving $m - r$ values when $v = 1$. Without loss of generality, suppose that the remaining values are $1, 2, \ldots, m - r$. Since the constraint $x_0 - x_1$ is flawless, note that there is a permutation π_0 such that each pair $(x_0 = i, x_1 = \pi_0(i))$ is allowed by the constraint. This means that all the pairs $(x_0 = 1, x_1 = \pi_0(1))$, $(x_0 = 2, x_1 = \pi_0(2))$, $\ldots$ $(x_0 = m-r, x_1 = \pi_0(m-r))$, are consistent with the constraint $x_0 - x_1$, so there are at least $m - r$ distinct values of x_1 consistent with the constraints in the chain $v - x_0 - x_1$ when $v = 1$. The process iterates since each constraint is flawless. Thus there are $m-r$ distinct values of x_k consistent with the chain of constraints $v - x_0 - x_1 - \ldots - x_k$ when $v = 1$. We have ignored only the constraint $x_k - v$. If this is to rule out $v = 1$, there must be at least $m - r$ conflicts involving $v = 1$. Thus between the two constraints $x_k - v$ and $v - x_0$, there are at least $r + (m - r) = m$ conflicts involving $v = 1$. Exactly the same holds for all m values of v, and the set of m conflicts that must exist for each value are all disjoint. Therefore the two constraints $x_k - v$ and $v - x_0$ contain at least m^2 conflicts, completing the proof.

Theorem 2. *In Models B or C with $p_1 = 2c/n$ and $c < 1/2$, almost all constraint graphs have no components with more than one cycle, that is the components are trees or unicyclic.*

Proof. For model C, the result follows from Theorem 4.2.6 from Palmer [46]. The same result applies to Model B, because the property of being a tree or unicyclic is 'convex' in the terms of Appendix VI of Palmer. Theorem 6.1 then applies.[2]

The following results follows immediately from Theorems 1 and 2.

Theorem 3. *Almost all random binary CSPs from models B or C with flawless constraints, $p_2 < \frac{1}{2}$, and fewer than cn constraints, for $c < 1/2$, are soluble.*

Two corollaries follow from this result which confirm the theoretical benefits of flawless problem generation that we have already claimed. The second is not immediately obvious, but follows because instances from the standard models B and C with $p_2 < \frac{1}{m}$ are automatically flawless. This can be shown by induction on the domain size m. The base case is that a 1×1 conflict matrix with $p_2 < 1$ must consist of a single, allowed, pair. In the step case, an $m \times m$ conflict matrix with $p_2 < \frac{1}{m}$ must have $m - 1$ or fewer conflicts. Suppose one conflict excludes the pair of values i, j. There must be at least one value j' consistent with i, so set $\pi(i) = j'$. Removing the row i and the column j' from the matrix yields an $m - 1 \times m - 1$ conflict matrix with $m - 2$ or fewer conflicts, and we can appeal to induction to complete the construction of the permutation π required for flawlessness.

Corollary 1. *Problems generated according to flawless model B or C at any value of $p_2 < \frac{1}{2}$ do not suffer asymptotically from trivial insolubility.*

Corollary 2. *Problems generated according to standard model B or C at any value of $p_2 < 1/m$ do not suffer asymptotically from trivial insolubility.*

These results do not apply directly to flawless models A and D, because for any value of $p_2 > 0$ they can generate individual conflict matrices with at least half of the possible conflicts selected. We can obtain similar asymptotic results if we condition the models to reject any such conflict matrix. While inelegant, this step will have little practical effect on generated problems where p_2 is significantly less than $1/2$. Apart from the rarity of matrices with the selected proportion of conflicts $\geq 1/2$, the proof of Theorem 1 shows that *every* pair of constraints in a cycle of flawless constraints must contain at least m^2 conflicts to make a cycle insoluble. This makes cyclic flaws even less likely in

[2] We thank Joe Culberson for help with this proof.

flawless models A and D. It is probable therefore that flawless models A and D will not be affected by flaws at practical problem sizes, and this likelihood can be extended to a guarantee if the models are adapted to ensure that the proportion of conflicts in any matrix $< 1/2$.

Achlioptas *et al.* [1] say that "Attempting to fix the old models, simply by conditioning on each value having degree less than D [m in our notation] in C [i.e. $p_2 < 1/m$] will probably not lead to any interesting new models." Instead, they suggest that it is more important to "shift from constraints that contain an entirely random subset of $p_2 D^2$ forbidden pairs [$p_2 m^2$ nogoods in our notation] ... to constraints where this subset has some structure." This is exactly what we have done by introducing flawless models. We suggest that flawless problem generation is a remedy more in keeping with Achlioptas *et al.*'s recommendation than their own model E. To guarantee an absence of flawed values, the minimum property required is arc consistency. However, this is not enough to prevent problems from being trivially insoluble. Our flawless generation method enforces a stronger condition than arc consistency, to guard against trivial insolubility for more complex reasons. Even if conflict matrices are arc consistent, if they are not flawless, a chain reaction of value removals can be triggered as propagation takes place. For example in the matrix of Figure 1 removing the single value 3 from either variable removes support from two values 1 & 2 of the other variable. Flawlessness prevents such chain reactions occurring and, as we have proved, prevents trivial asymptotic behaviour. It would be very interesting in the future to investigate the use of more complex structures in conflict matrices.

Our theoretical results for models B and C show a region of almost sure solubility when $p_1 < 1/(n-1)$. Problems in this region can contain many more conflicts than model E problems in their almost surely soluble region. That is because model E just adds one conflict at a time, while in our case each constraint can have up to $m^2/2 - 1$ conflicts in the flawless case. For models B, C, and the restricted versions of A and D, we have shown a soluble region when CSPs have $O(n)$ constraints, the same result obtained for model E previously [1]. When p_2 is fixed it is easy to show insolubility also occurs with $O(n)$ constraints. It is therefore likely that in these cases, there are genuine phase transitions between the almost-all-soluble and almost-all-insoluble regions.

To summarise, we have shown two surprising results. First, somewhat contrary to expectation, the value $p_2 = 1/m$ *does* precisely characterise the region of trivial insolubility in models B and C of binary CSPs. Second, for flawless generation methods, trivial insolubility is avoided up until the very high value of $p_2 = 1/2$. This second result can be made to apply to flawless models A and D if the models are adapted to reject constraints with at least half the possible conflicts selected.

9 Experimental Comparison of Flawless and Flawed Models

Based on our analysis in Section 6, the experiments from the literature most likely to contain flawed variables are those with domain size 3, such as those reported by Frost and Dechter in [18]. We would expect ordinary and flawless versions of model B to behave very differently on such problems. To test this, we implemented flawless model B and tested it against model B on the class $\langle 100, 3, p_1, 4/9 \rangle$. For this problem class, Frost and Dechter reported that 50% of problems were soluble in model B at 92 constraints, i.e. $p_1 \approx 0.01858$. We found a similar result, with 49.8% soluble problems at 92 constraints in model B with a sample of 1000. However, when the same parameters were used with flawless model B, we observed 99.3% solubility. This suggests that flawed variables played a significant role in this experiment. To confirm this, we generated and solved random problems using both models, varying the number of constraints (and hence p_1) to cover the transition from soluble to insoluble problems, with a sample size of 1000. The probability of solubility is shown in Figure 3. The transitions in solubility are very different for the two models. Indeed, the mushy region for model B (the region in which we have both soluble and insoluble problems) started with only 20 constraints on the 100 variables, at $p_1 \approx 0.004$, with a single insoluble problem. This problem contained a flawed variable. For flawless model B, we saw 52.4% solubility at 112 constraints ($p_1 \approx 0.023$) compared to 12.6% for model B. The transition for flawless model B is very sharp, whilst that for ordinary model B is very spread out. While these two transitions end at about the same place, the transitions may occur over completely different regions as n increases, with the flawed transition eventually converging on $p_1 = 0$.

Figure 4 shows the difference in median search cost, measured by the number of consistency checks. The problems were solved using the forward checking algorithm with conflict-directed backjumping and the fail-first heuristic (FC-CBJ-FF); the same algorithm was used for all subsequent experiments. The peak median cost for flawless problems is greater than for standard model B problems, and the flawless problems remain much harder as problems become more constrained. We conjecture that as problems become more constrained, flaws in flaw-prone problems become more common, and flawed problems will usually be quickly proved insoluble. The relative behaviour of other measures of search cost such as mean and maximum is broadly similar to that of the median.

Would flawless problem generation have affected experiments which were *not* influenced by flawed variables? To investigate this, we compared flawless model B with model B using the parameters $n = 20$ and $m = 10$ and sample size 1000 at each value of p_2. Results are shown in Figure 5. The transitions in probability are almost indistinguishable. The search cost is shown in Figure 6. Over the phase transition region, search cost is very similar in the two models. As problems become more constrained, flawless model B problems are very slightly harder to prove insoluble than ordinary model B problems. This is perhaps to be expected as flaws become more likely with increasing constrainedness.

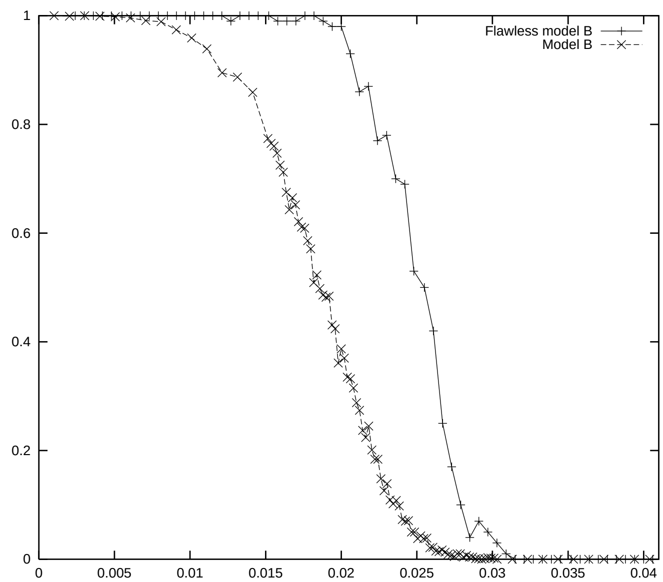

Fig. 3. Probability of solubility (y-axis) against p_1 (x-axis) for ordinary and flawless model B for $\langle 100, 3, p_1, 4/9 \rangle$ problems.

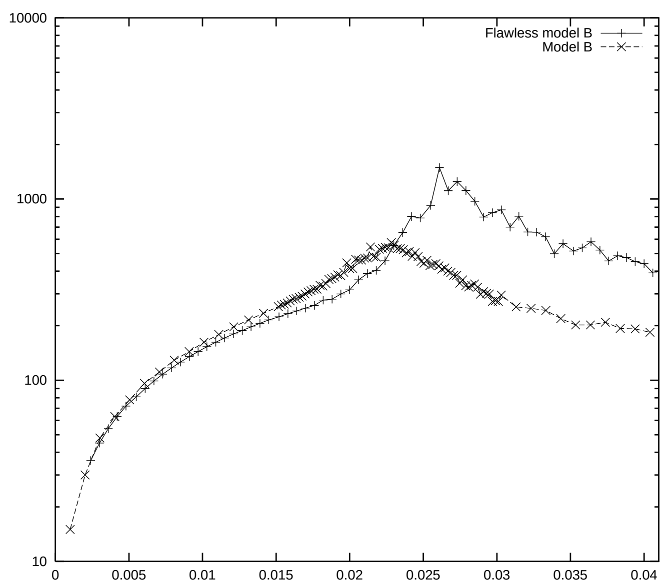

Fig. 4. Median number of consistency checks used (y-axis) against p_1 (x-axis) for FC-CBJ-FF on ordinary and flawless model B problems with $\langle 100, 3, p_1, 4/9 \rangle$.

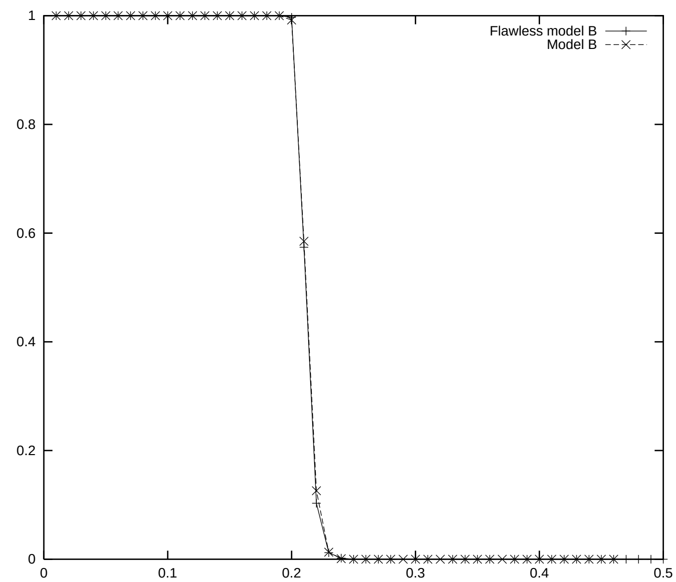

Fig. 5. Probability of solubility (y-axis) against p_2 (x-axis) for ordinary and flawless $\langle 20, 10, 1, p_2 \rangle$ model B problems.

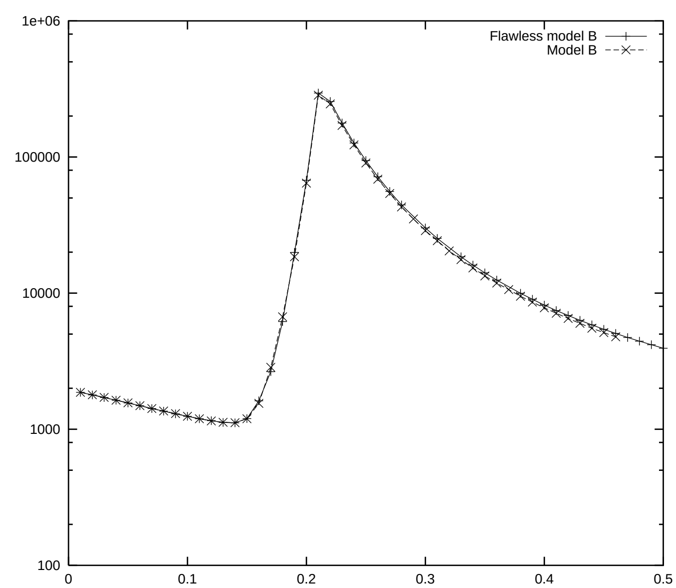

Fig. 6. Median number of checks used (y-axis) against p_2 (x-axis) for ordinary and flawless $\langle 20, 10, 1, p_2 \rangle$ model B problems.

To conclude, experimenters should be aware of the danger of producing problems with flawed values when using the standard models A to D, and should use a flawless generator when the occurrence of flaws might affect their conclusions. For many purposes, a flawless generator could be used as a matter of course, since the results will only be significantly different from those produced by the equivalent flaw-prone generators exactly when flaws are occurring.

10 Structured Constraint Graphs

Random problems provide a plentiful and unbiased source of problems for benchmarking. However, we must be careful that our algorithms do not become tuned to solve random problems and perform poorly on real problems. Real problems can contain structures that occur very rarely in the models discussed here, even when the real problems contain only binary constraints. For example, in a graph colouring problem derived from a 1994 exam time-tabling problem at Edinburgh University, Gent and Walsh found a 10-clique of nodes with only 9 colours available [30]. This was in a 59 node graph with 485 edges. The presence of this clique dominated the performance of their graph colouring algorithm.

Random graphs of similar size and density are very unlikely to contain such a large clique. The probability that k given nodes in a random graph with n nodes and e edges are connected by the right edges to form a k-clique is,

$$\prod_{i=0}^{\binom{k}{2}-1} \frac{e-i}{\binom{n}{2}-i}$$

From this we can get the expected number of k-cliques and hence by the Markov inequality a bound on the probability of the graph containing a k-clique:

$$\Pr\{m\text{-clique in graph of } n \text{ nodes \& } e \text{ edges}\} \leq \binom{n}{k} \prod_{i=0}^{\binom{k}{2}-1} \frac{e-i}{\binom{n}{2}-i}$$

For $n = 59$, $k = 10$ and $e = 485$, the probability of clique of size 10 or larger is less than 10^{-14}. It is thus very unlikely that a random graph of the same size and density as the graph in the exam time-tabling problem would contain a regular structure like a 10-clique. However, cliques of this size occur in the real data due to the module structure within courses.

As another example, Gomes *et al.* have proposed quasigroup completion as a constraint satisfaction benchmark that models some of the structure found in sports scheduling and fibre-optic routing problems [32]. Quasigroup completion is the problem of filling in the missing entries in a Latin square, a multiplication table in which each entry appears once in every row and column. An order n quasigroup problem can be formulated as n-colouring a graph with n^2 nodes and $n^2(n-1)$ edges. The edges form $2n$ cliques, with each clique being of size

n and representing the constraint that each colour appears once in every row or column. For example, an order 10 quasigroup has 20 cliques of size 10 in a 100 node graph with 900 edges. With a random graph of this size and edge density, the probability of a clique of size 10 or larger is less than 10^{-20}. It is thus extremely unlikely that a random graph of this size and density would contain a regular structure like a 10-clique, let alone 20 of them linked together. The random models will therefore not generate sets of problems like the exam time-tabling problem or quasigroup completion.

It is clear that the existing models can be adapted to use any constraint graph that is desired. In models A to D, problems are generated in a two stage process: the first stage is to generate a constraint graph, and the second stage is to generate conflict matrices for edges in this graph. There is no technical reason why the first stage must be random; it can instead involve a particular constraint graph. This approach was taken by Smith and Grant [53] where they used a "braided" constraint graph and generated random constraints on this as in model B. Given a particular constraint graph, we can then generate the conflict matrices as in models A & C or as in models B & D, including generating flawless constraints if required.

To determine how these structured models differ from unstructured models, we experimented on the timetabling and quasigroup graphs mentioned earlier. To focus the comparison just on the introduction of structure into the constraint graph, we only report results for flawless model B. We observed broadly similar results using model B. Our first experiments are on the constraint graph taken from the quasigroup problem of order 7. This problem has 49 variables all with domains of size 7. Each variable is in constraints with 12 others, giving a total of 294 constraints. In the original quasigroup problem these constraints are difference constraints. Here, we randomly generated flawless constraints of different tightnesses using the model B method. As a comparison, we generated unstructured flawless model B problems with the same number of variables, same domain size and same density of edges in their constraint graph, i.e. $\langle 49, 7, 0.25, p_2 \rangle$ problems. We tested 100 problems at each value of p_2 from 1/49 to 25/49 in steps of 1/49, using the FC-CBJ-FF algorithm. Results are shown in Figure 7 and Figure 8. While the transition in solubility occurs at very similar values of p_2 in the two experiments, there is a large difference in search cost. In particular, the structured instances seem much harder than the random problems at the phase transition.

We also experimented with the constraint graph derived from the 1994 exam timetabling problem at Edinburgh University. The graph has 59 nodes and 485 edges. Nodes correspond to exams, while each edge corresponds to two exams to be taken by one student, for which clashes must be avoided. In the original problem there were 36 values, corresponding to 9 days with 4 exams per day. Gent and Walsh solved the original problem using using Prosser's CSPLab code for FC-CBJ-FF with directed k-consistency [30]. The problem was insoluble and took 411,770,462 consistency checks. Unfortunately, solving an ensemble of structured problems based on this constraint graph was prohibitively expensive

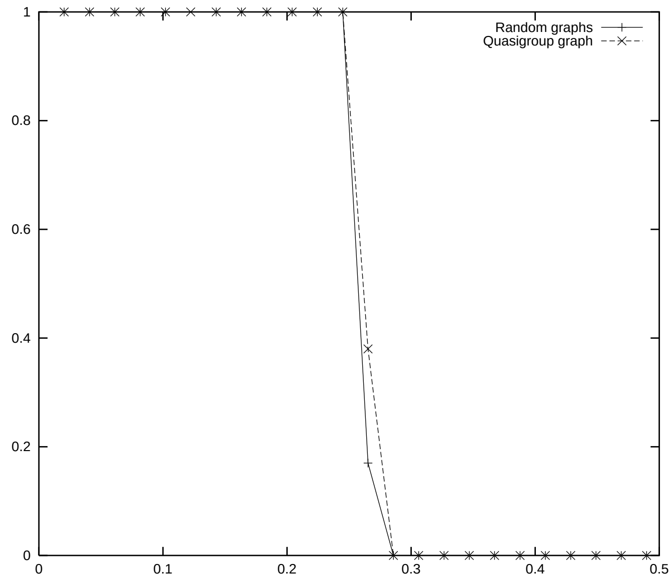

Fig. 7. Probability of solubility (y-axis) against p_2 (x-axis) for flawless model B problems generated with either the constraint graph of a quasigroup problem of order 7 or a random constraint graphs with the same number of nodes and edges.

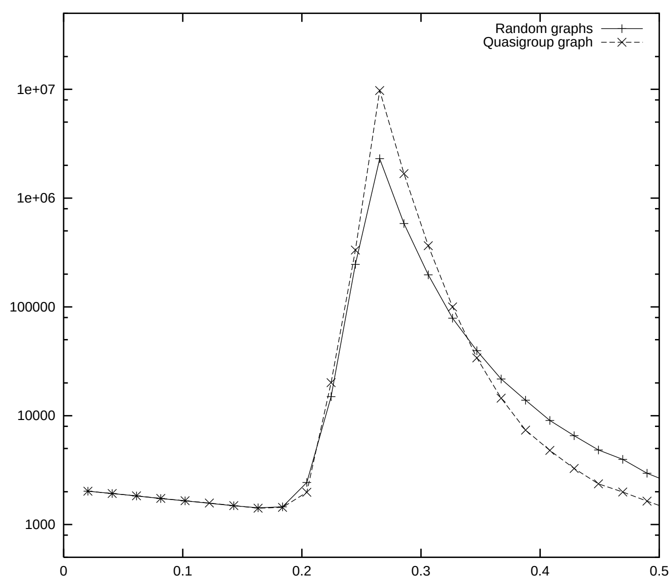

Fig. 8. Median number of checks used (y-axis) against p_2 (x-axis) for flawless model B problems generated with either the constraint graph of a quasigroup problem of order 7 or a random constraint graphs with the same number of nodes and edges.

with 36 values in the domain of each variable. Gent and Walsh showed that the original exam timetabling problem was insoluble because it contains a 10-clique of exams which all had to happen at different times with only 9 time slots available. Since our problem generation method preserves this 10-clique, we generated problems with 9 values for each variable. We tested flawless model B on random problems generated either with this constraint graph, or with a random graph with the same number of nodes and edges. Sample size was again 100, and we tested values of p_2 from 1/81 to 35/81 in steps of 1/81 using FC-CBJ-FF.

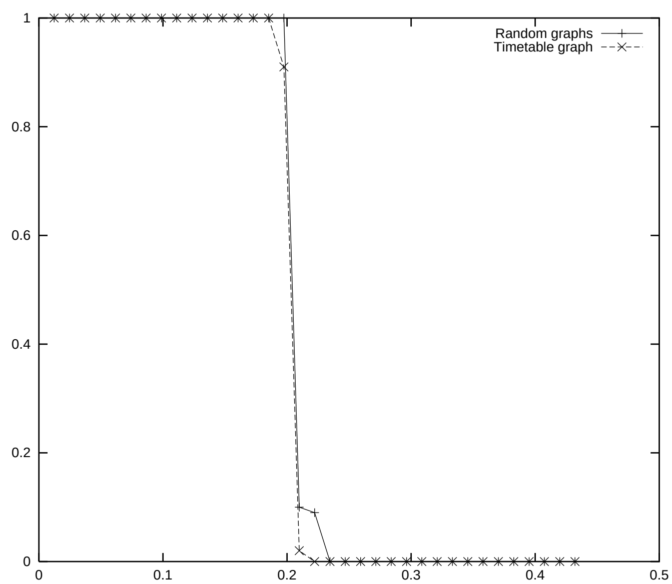

Fig. 9. Probability of solubility (y-axis) against p_2 (x-axis) for flawless model B problems generated with either the constraint graph of the 1994 exam timetabling problem or a random constraint graph with the same number of nodes and edges.

Figure 9 shows the probability in solubility as the constraint tightness is varied. The transition in solubility for problems with random constraint graphs is almost identical to that for problems with structured constraint graphs. Figure 10 shows the median search cost. The difference in search cost is the opposite of that seen with the quasigroup constraint graph. Problems with random constraint graphs require about 100 times more consistency checks at and beyond $p_2 = 0.21$. Similar behaviour is seen in mean and maximum search cost.

To summarise, we have experimented with ensembles of problems based on specific constraint graphs. Such structured problem generation is particularly interesting when the constraint graph contains structure unlikely to occur in random graphs. We have experimented on two such graphs, based on a quasigroup and a timetabling problem. In both cases, search cost was very different

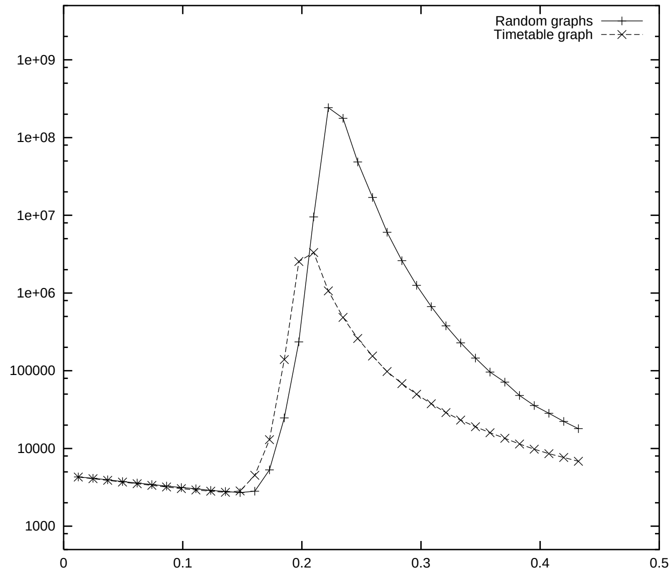

Fig. 10. Median number of checks used (y-axis) against p_2 (x-axis) for flawless model B problems generated with either the constraint graph of the 1994 exam timetabling problem or a random constraint graph with the same number of nodes and edges.

to that seen with existing random models; structured problems using the quasigroup constraint graph were harder than purely random problems with equivalent parameter values, while the timetabling graph gave easier problems than the random problems. Structured problem generation allows us to repeatedly test constraint graphs of special interest. This helps address the difficulty that randomly generated problems may not be realistic, whilst realistic problems may be hard to collect in statistically significant sample sizes.

11 Conclusions

We have performed a detailed study of the consequences of a recent theoretical result of Achlioptas *et al.* [1]. This result shows that as the number of varibles increases, the traditional models of random problems almost surely contain a flawed variable and are therefore trivially insoluble, provided the constraint tightness is at least $1/m$, where m is the domain size. We proved that this result is tight for models B and C since they do not suffer from such flaws for $p_2 < 1/m$. Our survey of previous experimental studies shows that many studies have, however, used problems with $p_2 \geq 1/m$. Fortunately, most (but not all) of these studies use too few variables and too large domains to contain flawed variables. As expected, flawed variables occur most often with dense constraint graphs and small domains. With constraint graphs of fixed average degree and large domains, the possibility of flawed variables can usually be ignored.

Achlioptas *et al.* propose an alternative random model (model E) which does not suffer from the deficiencies of the standard models as the number of variables increases, and so give the first evidence that there could be an asymptotic phase transition in random constraint satisfaction problems. However, from the experimental point of view, model E is much less flexible than the standard models, since the constraint density and constraint tightness cannot be controlled independently.

We have shown how a limited amount of structure can be introduced into the conflict matrices to make them flawless. We have proved that problems generated by flawless variants of the models A, B, C and D are not trivially insoluble in the limit for all values of p_2 less than $1/2$. We can thereby generate ensembles of problems that are not trivially insoluble due to the presence of flawed variables. We have also reported on experiments with problems that contain structures in their constraint graphs which are rare in random graphs.

What general lessons can be learnt from this study? First, experiments can benefit greatly from theoretical results like those of Achlioptas *et al.* Flawed variables are likely to have occurred in a small but significant number of previous experimental studies. A simple arc consistency algorithm would therefore have shown very quickly that these problems have no solution. Experimenters should take this into account when planning future experiments, and consider choosing a flawless problem generator. Second, theory can benefit greatly from experiments. Theory provided estimates for the probability of problems having flawed variables based on some simplifying assumptions. Experiments quickly determined the accuracy of such estimates. Third, we must continue to improve and extend our random models so that we have a wide range of realistic and hard problems on which to test algorithms. Such extensions can introduce structure either into the constraint graph (as in the experiments reported in section 10) or into the conflict matrix (as in the flawless generation method proposed here) or both.

References

1. D. Achlioptas, L.M. Kirousis, E. Kranakis, D. Krizanc, M.S.O. Molloy, and Y.C. Stamatiou. Random constraint satisfaction: A more accurate picture. In *Proc. CP97*, pages 107–120. Springer, 1997.
2. F. Bacchus and A. Grove. On the forward checking algorithm. In U. Montanari and F. Rossi, editors, *Principles and Practice of Constraint Programming*, pages 292–309. Springer, 1995.
3. F. Bacchus and P. van Run. Dynamic variable ordering in CSP's. In U. Montanari and F. Rossi, editors, *Principles and Practice of Constraint Programming*, pages 258–275. Springer, 1995.
4. C. Bessière and J-C. Régin. MAC and combined heuristics: Two reasons to forsake FC (and CBJ?) on hard problems. In E.C. Freuder, editor, *Proceedings of Second International Conference on Principles and Practice of Constraint Programming (CP96)*, pages 61–75. Springer, 1996.

5. J.E. Borrett, E. Tsang, and N.R. Walsh. Adaptive constraint satisfaction: The quickest first principle. In W. Wahlster, editor, *Proceedings of the 12th ECAI*, pages 160–164. European Conference on Artificial Intelligence, Wiley, 1996.
6. B. Cabon, G. Verfaillie, D. Martinez, and P. Bourret. Using mean field methods for boosting backtrack search in constraint satisfaction problems. In W. Wahlster, editor, *Proceedings of the 12th ECAI*, pages 165–169. European Conference on Artificial Intelligence, Wiley, 1996.
7. P. Cheeseman, B. Kanefsky, and W.M. Taylor. Where the really hard problems are. In *Proceedings of the 12th IJCAI*, pages 331–337, 1991.
8. A. Chmeiss and P. Jégou. Path-consistency: when space misses time. In *Proceedings of AAAI-96*, pages 196–201, 1996.
9. D.A. Clark, J. Frank, I.P. Gent, E. MacIntyre, N. Tomov, and T. Walsh. Local search and the number of solutions. In *Proceedings of CP-96*, pages 119–133. Springer, 1996.
10. R. Debruyne and C. Bessière. From restricted path consistency to max-restricted path consistency. In G. Smolka, editor, *Proceedings of Third International Conference on Principles and Practice of Constraint Programming (CP97)*, pages 312–326. Springer, 1997.
11. R. Debruyne and C. Bessière. Some practicable filtering techniques for the constraint satisfaction problem. In *Proceedings of the 15th IJCAI*, pages 412–417. International Joint Conference on Artificial Intelligence, 1997.
12. E. Freuder. A sufficient condition for backtrack-free search. *Journal of the Association for Computing Machinery*, 29(1):24–32, 1982.
13. E.C. Freuder and C.D. Elfe. Neighbourhood inverse consistency preprocessing. In *Proceedings of AAAI-96*, pages 202–208, 1996.
14. E.C. Freuder and P.D. Hubbe. Extracting constraint satisfaction subproblems. In *Proceedings of the 14th IJCAI*, pages 548–555. International Joint Conference on Artificial Intelligence, 1995.
15. E. Friedgut. Sharp thresholds for graph properties and the k-SAT problem, 1998. Unpublished manuscript.
16. A. Frieze and S. Suen. Analysis of two simple heuristics on a random instance of k-SAT. *Journal of Algorithms*, 20:312–355, 1996.
17. D. Frost and R. Dechter. Dead-end driven learning. In *Proceedings AAAI-94*, pages 294–300, 1994.
18. D. Frost and R. Dechter. In search of the best search: an empirical evaluation. In *Proceedings AAAI-94*, pages 301–306, 1994.
19. D. Frost and R. Dechter. Look-ahead value ordering for constraint satisfaction problems. In *Proceedings of the 14th IJCAI*, pages 572–578. International Joint Conference on Artificial Intelligence, 1995.
20. D. Frost, I. Rish, and L. Vila. Summarizing CSP hardness with continuous probability distributions. In *Proceedings of the 14th National Conference on AI*, pages 327–333. American Association for Artificial Intelligence, 1997.
21. P. Galinier and J-K. Hao. Tabu search for maximal constraint satisfaction problems. In G. Smolka, editor, *Proceedings of Third International Conference on Principles and Practice of Constraint Programming (CP97)*, pages 196–208. Springer, 1997.
22. D.R. Gaur, W.K. Jackson, and W.S. Havens. Detecting unsatisfiable CSPs by coloring the micro-structure. In *Proceedings of the 14th National Conference on AI*, pages 215–220. American Association for Artificial Intelligence, 1997.
23. I.P. Gent, E. MacIntyre, P. Prosser, P. Shaw, and T. Walsh. The constrainedness of arc consistency. In *Proceedings of CP-97*, pages 327–340. Springer, 1997.

24. I.P. Gent, E. MacIntyre, P. Prosser, B.M. Smith, and T. Walsh. An empirical study of dynamic variable ordering heuristics for the constraint satisfaction problem. In *Proceedings of CP-96*, pages 179–193. Springer, 1996.
25. I.P. Gent, E. MacIntyre, P. Prosser, B.M. Smith, and T. Walsh. Random constraint satisfaction: Flaws and structure. Technical Report APES-08-1998, APES Research Group, 1998.
26. I.P. Gent, E. MacIntyre, P. Prosser, and T. Walsh. Scaling effects in the CSP phase transition. In U. Montanari and F. Rossi, editors, *Principles and Practice of Constraint Programming*, pages 70–87. Springer, 1995.
27. I.P. Gent, E. MacIntyre, P. Prosser, and T. Walsh. The constrainedness of search. In *Proceedings of AAAI-96*, pages 246–252, 1996.
28. I.P. Gent, E. MacIntyre, P. Prosser, and T. Walsh. The scaling of search cost. In *Proceedings of AAAI-97*, pages 315–320, 1997.
29. I.P. Gent and J. Underwood. The logic of search algorithms: Theory and applications. In *Principles and Practice of Constraint Programming – CP97*, pages 77–91. Springer, 1997.
30. I.P. Gent and T. Walsh. Phase transitions from real computational problems. In *Proceedings of the 8th International Symposium on Artificial Intelligence*, pages 356–364, 1995.
31. I.P. Gent and T. Walsh. Analysis of heuristics for number partitioning. *Computational Intelligence*, 1998. In press.
32. C. Gomes and B. Selman. Problem structure in the presence of perturbations. In *Proceedings of the 14th National Conference on AI*, pages 221–226. American Association for Artificial Intelligence, 1997.
33. S. Grant and B.M. Smith. The phase transition behaviour of maintaining arc consistency. Research Report 95.25, School of Computer Studies, University of Leeds, 1995. A revised and shortened version appears in Proceedings of 12th ECAI, pages 175–179, 1996.
34. S.A. Grant and B.M. Smith. The phase transition behaviour of maintaining arc consistency. In *Proceedings of ECAI-96*, pages 175–179, 1996.
35. K. Kask and R. Dechter. Gsat and local consistency. In *Proceedings of the 14th IJCAI*, pages 616–622. International Joint Conference on Artificial Intelligence, 1995.
36. K. Kask and R. Dechter. A graph-based method for improving GSAT. In *Proceedings of AAAI-96*, pages 350–355, 1996.
37. L.M. Kirousis, E. Kranakis, and D. Krizanc. Approximating the unsatisfiability threshold of random formulas. In *Proceedings of the 4th Annual European Symposium on Algorithms (ESA'96)*, pages 27–38, 1996.
38. A.C.M. Kwan, E.P.K. Tsang, and J.E. Borrett. Predicting phase transitions of binary CSPs with constraint graph information. In W. Wahlster, editor, *Proceedings of the 12th ECAI*, pages 185–189. European Conference on Artificial Intelligence, Wiley, 1996.
39. J. Larrosa and P. Meseguer. Optimization-based heuristics for maximal constraint satisfaction. In *Proceedings of Second International Conference on Principles and Practice of Constraint Programming (CP95)*, pages 103–120. Springer, 1995.
40. J. Larrosa and P. Meseguer. Exploiting the use of DAC in MAX-CSP. In E.C. Freuder, editor, *Proceedings of Second International Conference on Principles and Practice of Constraint Programming (CP96)*, pages 308–322. Springer, 1996.
41. J. Larrosa and P. Meseguer. Phase transition in MAX-CSP. In W. Wahlster, editor, *Proceedings of the 12th ECAI*, pages 190–194. European Conference on Artificial Intelligence, Wiley, 1996.

42. D. Lesaint. Maximal sets of solutions for constraint satisfaction problems. In *Proceedings of ECAI-94*, pages 110–114, 1994.
43. A. Meisels, S.E. Shimony, and G. Solotorevsky. Bayes networks for estimating the number of solutions to a CSP. In *Proceedings of the 14th National Conference on AI*, pages 179–184. American Association for Artificial Intelligence, 1997.
44. P. Meseguer and J. Larrosa. Constraint satisfaction as global optimization. In *Proceedings of the 14th IJCAI*, pages 579–584. International Joint Conference on Artificial Intelligence, 1995.
45. D. Mitchell, B. Selman, and H. Levesque. Hard and easy distributions of SAT problems. In *Proceedings of AAAI-92*, pages 459–465, 1992.
46. E.M. Palmer. *Graphical Evolution: An introduction to the Theory of Random Graphs*. John Wiley & Sons, 1985.
47. P. Prosser. Binary constraint satisfaction problems: Some are harder than others. In *Proceedings of ECAI-94*, pages 95–99, 1994.
48. I. Rish and D. Frost. Statistical analysis of backtracking on inconsistent CSPs. In G. Smolka, editor, *Proceedings of Third International Conference on Principles and Practice of Constraint Programming (CP97)*, pages 150–162. Springer, 1997.
49. D. Sabin and E.C. Freuder. Contradicting conventional wisdom in constraint satisfaction. In *Proceedings of ECAI-94*, pages 125–129, 1994.
50. D. Sabin and E.C. Freuder. Understanding and improving the mac algorithm. In G. Smolka, editor, *Proceedings of Third International Conference on Principles and Practice of Constraint Programming (CP97)*, pages 167–181. Springer, 1997.
51. B.M Smith. Phase transition and the mushy region in constraint satisfaction problems. In *Proceedings of ECAI-94*, pages 100–104, 1994.
52. B.M. Smith and M.E. Dyer. Locating the phase transition in binary constraint satisfaction problems. *Artificial Intelligence*, 81:155–181, 1996.
53. B.M. Smith and S. Grant. Where the *exceptionally* hard problems are. In *Proc. CP-95 Workshop on Studying and Solving Really Hard Problems*, pages 172–182, 1995.
54. B.M. Smith and S. Grant. Modelling exceptionally hard constraint satisfaction problems. In G. Smolka, editor, *Proceedings of Third International Conference on Principles and Practice of Constraint Programming (CP97)*, pages 182–195. Springer, 1997.
55. B.M. Smith and S.A. Grant. Sparse constraint graphs and exceptionally hard problems. In *Proceedings of IJCAI-95*, pages 646–651, 1995.
56. B.M. Smith. Constructing an Asymptotic Phase Transition in Random Binary Constraint Satisfaction Problems. To appear in *Journal of Theoretical Computer Science, Special Issue on NP-Hardness and Phase Transitions.*
57. R.J. Wallace. Analysis of heuristic methods for partial constraint satisfaction problems. In E.C. Freuder, editor, *Proceedings of Second International Conference on Principles and Practice of Constraint Programming (CP96)*, pages 482–496. Springer, 1996.
58. C. Williams and T. Hogg. Exploiting the deep structure of constraint problems. *Artificial Intelligence*, 70:73–117, 1994.
59. K. Xu and W. Li. Exact Phase Transitions in Random Constraint Satisfaction Problems. *Journal of AI Research*, 12:93–103, 2000.
60. N. Yugami, Y. Ohta, and H. Hara. Improving repair-based constraint satisfaction methods by value propagation. In *Proceedings AAAI-94*, pages 344–349, 1994.