Random Constraint Satisfaction: A More Accurate Picture

Dimitris Achlioptas,[1⋆] Lefteris M. Kirousis,[2⋆⋆] Evangelos Kranakis,[3⋆⋆⋆]
Danny Krizanc,[3†] Michael S.O. Molloy,[1‡] and Yannis C. Stamatiou[2§]

[1] Department of Computer Science, University of Toronto, Toronto, ON M5S 3G4, Canada
[2] Department of Computer Engineering and Informatics, University of Patras, Rio, 26500 Patras, Greece
[3] School of Computer Science, Carleton University, Ottawa, ON K1S 5B6, Canada

July 29, 1998

Abstract. In the last few years there has been a great amount of interest in Random Constraint Satisfaction Problems, both from an experimental and a theoretical point of view. Quite intriguingly, experimental results with various models for generating random CSP instances suggest that the probability of such problems having a solution exhibits a "threshold-like" behaviour. In this spirit, some preliminary theoretical work has also been done in analyzing these models asymptotically, i.e. as the number of variables becomes large. In this paper we prove that, contrary to beliefs based on experimental evidence, the models commonly used for generating random CSP instances *do not* have an asymptotic threshold. In particular, we prove that asymptotically *almost all* instances they generate are overconstrained, sufferring from trivial, local inconsistencies. To complement this result we present an alternative, single-parameter model for generating random CSP instances and prove that, unlike current models, it is non-trivial asymptotically. Moreover, we derive explicit bounds for the narrow region within which the probability of having a solution changes dramatically.

1 Introduction

A *constraint network* comprises n variables, with their respective domains, and a set of *constraint relations* each binding a subset of the variables (for a def-

⋆ Supported by a Natural Sciences and Engineering Research Council (NSERC) of Canada PGS Scholarship. E-mail: optas@cs.toronto.edu
⋆⋆ Partially supported by the EU ESPRIT Long-term Research Project ALCOM-IT (Project Nr. 20244). E-mail: kirousis@ceid.upatras.gr
⋆⋆⋆ Supported in part by an NSERC grant. E-mail: kranakis@scs.carleton.ca
† Supported in part by an NSERC grant. E-mail: krizanc@scs.carleton.ca
‡ Supported in part by an NSERC grant. E-mail: molloy@cs.toronto.edu
§ Partially supported by the EU ESPRIT Long-term Research Project ALCOM-IT (Project Nr. 20244) and the General Secretariat of Research and Technology of Greece (Project YPER 1248). E-mail: stamatiu@ceid.upatras.gr

inition see Section 2). Given a constraint network, the *Constraint Satisfaction Problem* (CSP) is that of determining the n-tuples of value assignments that are compatible with all the constraints. CSP is a fundamental problem in Artificial Intelligence, with numerous applications ranging from scene labeling to scheduling and knowledge representation [10, 22, 28].

Following the seminal work of Cheeseman, Kanefsky and Taylor [8], there has been a growing interest in the study of the relation between the parameters that define an instance of CSP (i.e., number of variables, domain size, tightness of constraints etc.) and: (i) the likelihood that the instance has a solution, (ii) the difficulty with which such a solution can be found. An extensive account of relevant results, both experimental and theoretical, appears in [3]. Perhaps the most commonly used practice for conducting experiments with CSP is to generate a large set of random instances, all with the same defining parameters, and then for each instance in the set to use heuristic methods (as, in general, CSP is NP-complete) for deciding whether a solution exists. The proportion of instances that have a solution is used as an indication of the "likelihood", while the average time taken, per instance, captures a notion of "hardness" for such instances.

Research in this direction originated from research on random instances not of general CSP, but of specific problems like SAT and graph colorability (see [8]). Especially, for random instances of SAT, Mitchell, Selman and Levesque ([24]) pointed out that some distributions that are commonly used in such experiments are uninteresting since they generate formulas that are almost always very easy to satisfy. On the other hand, they reported that the distribution where each formula has precisely k literals per clause, i.e., where it is an instance of k-SAT, can generate some very hard formulas for $k \geq 3$. In particular, they reported that for $k = 3$ the following remarkable behavior is observed in experiments: let r denote the ratio of clauses to variables. For $r < 4$ almost all formulas are satisfiable and a satisfying truth assignment is easy to find. For $r > 4.5$ almost all formulas are unsatisfiable. Finally, a "50%-solubility" point seems to appear around 4.2, the same point where the computational complexity of finding a satisfying truth assignment is maximized. Further experimental results, as well the connection of this threshold behavior to phase transition phenomena in physics, are described in [20].

The experimental results have motivated a theoretical interest in understanding the behavior of random k-SAT formulas, as the number of variables becomes asymptotic. In the following, we give a brief summary of the strongest known results. We will say that a sequence of events $\mathcal{E}_n$ occurs "almost certainly" if $\Pr[\mathcal{E}_n]$ tends to 1 as $n \to \infty$. For $k = 2$, a sharp threshold was proved in [9, 16]: a random instance of 2-SAT is almost certainly satisfiable if $r < 1$ and almost certainly unsatisfiable if $r > 1$. For $k = 3$, there has been a series of results [6, 7, 5, 13, 19, 15, 21, 11] narrowing the area for which we do not have almost certain (un)satisfiability. The best bounds currently known come from [15] where it was proven that a random instance of 3-SAT is almost certainly satisfiable if $r < 3.003$, and [21] where it was proven that a random instance of

3-SAT is almost certainly unsatisfiable if $r > 4.602$. For general k, the best known lower bound for r is $\Theta(2^k/k)$ [7, 9, 15] while 2^k is an easy upper bound and was improved by a constant factor in [21] by extending the techniques used for 3-SAT. Very recently, Friedgut [14] made great progress towards establishing the *existence* of a threshold for k-SAT for all values of k, yet his approach does not provide any information regarding the threshold's asymptotic location.

For general CSP, there has been accumulating *experimental* evidence that the parameters which define the distribution of random instances have a *critical region*. In particular, this is taken to be the region within which random instances appear to be much "harder" to solve, that is to determine whether they have any solution. On the other hand, random instances with defining parameters outside this region, seem to either almost certainly have a solution or to almost certainly not and, furthermore, finding a solution when one exists appears to be much easier. Finally, the "hard" region seems to narrow as the number of variables increases. This experimental evidence has led to the suggestion (often the claim) that a threshold phenomenon underlies Random CSP [8, 17, 25, 26, 29, 30, 31]. A first step towards an asymptotic analysis of Random CSP was made by Smith and Dyer in [26], who examined a well-established model for generating random CSP with binary constraints. Using the First Order Moment method they derive a condition on the defining parameters subject to which the probability that a random instance has a solution tends to 0 as $n \to \infty$. They also study the variance of the number of solutions as a function of the defining parameters.

In this paper we show that if the number of permissible pairs of value assignments in each constraint is a constant fraction of all possible pairs (as, e.g., is assumed in [26]), then as the number of variables increases, the probability that a solution exists tends to zero. Obviously, this implies that a threshold phenomenon *does not* underlie Random CSP generated using any such models, including those examined in [26], since asymptotically *almost all* instances generated will not have a solution. The threshold-like picture given by experimental studies is misleading. In particular, 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).

Roughly speaking, the reason for this deficiency of the standard models of general random CSP is the following. Since in each constraint, every pair of value assignments is forbidden *randomly and with constant probability*, a variable has *constant* positive probability that it cannot be set to any of its values so as to satisfy the constraints it is bound by, even if every other variable was free to take any value, i.e. the inconsistency is trivial, occuring for a completely "local" reason. Now, this constant positive probability of a local inconsistency for a fixed variable, will be shown to imply that almost certainly there exists at least one variable suffering from this problem. Let us emphasize that this phenomenon does not occur in "specific" CSP (such as SAT or colorability), where there is some structure, or even no randomness at all, in the forbidden pairs of values for two constrained variables. Thus, we conclude that a new model is necessary

for general random CSP before any asymptotic analysis can be performed.

We make a first step in this direction by proposing a simple, alternative model which has only one parameter r. We provide a preliminary analysis of the model's asymptotic behavior which proves that the instances generated to not have trivial local inconsistencies. Moreover, we show how to almost certainly find a solution for a random CSP instance if r is smaller than some upper bound and also we give a counting argument showing that no solution exists if r is greater than some other bound. The analysis for the underconstrained region exploits a connection to random graphs for binary constraints and an analysis adapted from [14] and [7] for arbitrary arity constraints larger than 2, while the counting argument is based on a technique introduced in [21].

2 Definitions and notation

A *constraint network* consists of a set of variables $X_1, \ldots, X_n$ with respective domains $D_1, \ldots, D_n$, and a set of constraints $\mathcal{C}$. For $2 \leq k \leq n$ a constraint $R_{i_1,i_2,\ldots,i_k} \in \mathcal{C}$ is a subset of $D_{i_1} \times D_{i_2} \cdots D_{i_k}$, where the $i_1, i_2, \ldots, i_k$ are distinct. We say that $R_{i_1,i_2,\ldots,i_k}$ is of arity k xand that it bounds the variables $X_{i_1}, \ldots, X_{i_k}$. For a given constraint network, the *Constraint Satisfaction Problem* (CSP), asks for all the n-tuples $(d_1, \ldots, d_n)$ of values such that $d_i \in D_i$, $i = 1, \ldots, n$, and for every $R_{i_1,i_2,\ldots,i_k} \in \mathcal{C}$, $(d_{i_1}, d_{i_2}, \ldots, d_{i_k}) \notin R_{i_1,i_2,\ldots,i_k}$. Such an n-tuple is called a *solution* of the CSP. The decision version of the CSP is determining whether a solution exists.

For an instance Π of CSP with n variables, its *constraint hypergraph* G^{Π} has n vertices $v_1, v_2, \ldots, v_n$, which correspond to the variables of Π and it contains a hyperedge $\{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$ iff there exists a constraint of arity k that bounds the variables $X_{i_1}, X_{i_2}, \ldots, X_{i_k}$. We will use the following convenient graph-theoretic representation of a CSP instance Π. The *incompatibility hypergraph* of Π, C^{Π} is an n-partite hypergraph. The ith part of C^{Π} corresponds to variable X_i of Π and it has exactly $|D_i|$ vertices, one for each value in D_i. In C^{Π} there exists a hyperedge $\{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$, iff the corresponding values $d_{i_1} \in D_{i_1}$, $d_{i_2} \in D_{i_2}$, $\ldots$, $d_{i_k} \in D_{i_k}$ are in (not allowed by) some constraint that bounds the corresponding variables. Often, for both the constraint and the incompatibility hypergraph we will omit the superscript if it is clear from the context. Hence, the decision version of CSP is equivalent to asking if there exists a set of vertices in C containing exactly one vertex from each part while not "containing" any hyperedge, i.e., whether there exists an independent set with one vertex from each part.

2.1 Currently used models for random CSP

In generating random CSP instances it is common practice to make the simplifying assumption that all the variable domains contain the same number of values $D \geq 2$. We will adhere to this convention as it simplifies considerably the technical manipulations in the paper, without having any significant effect

on the conclusions. Hence, fixing n and D, the generation of a random CSP is usually done in two steps. First, the constraint graph G (recall that $k = 2$) is constructed as a random graph and then for each edge (constraint) in G a set of edges (incompatible pairs of values) is inserted in C (the constraint graph). A general framework for doing this is presented in [25, 26]. More precisely,

Step 1. *Either (i) each one of the $\binom{n}{2}$ edges is selected to be in G independently of all other edges with probability p_1, or (ii) we uniformly select a random set of edges of size $p_1\binom{n}{2}$.*

Step 2. *Either (i) for every edge of G each one of the D^2 edges in C is selected with probability p_2, or (ii) for every edge of G we uniformly select a random set of edges in C of size $p_2 D^2$.*

The parameter p_1 determines how many constraints exist in a CSP instance and it is called *constraint density*, whereas p_2 determines how restrictive the constraints are and it is called *constraint tightness*. It is not hard to see that unless the total number of incompatible pairs of values is $\Theta(n)$ the resulting instances are underconstrained or overconstrained. The parameter $p_2 \in (0, 1)$ is taken to be a constant independent of any other model parameters. Since, p_2 is a constant and D, k are also bounded, p_1 is varied so that there are $\Theta(n)$ incompatible pairs of values in total. Combining the options for the two steps, we get four slightly different models for generating random CSP which have received various names in the past literature. In particular, in the terminology used in [26], if both Step 1 and Step 2 are done using the first option (the $G_{n,p}$ fashion, in random graphs terminology) then we get Model A, while if they are both done using the second option (the $G_{n,m}$ fashion, in random graphs terminology), we get Model B. Our criticism in the next section applies to all four variations.

3 Shortcomings of currently used models

Let Π be a random CSP with D or more incompatible pairs in each constraint (suppose that we use the second option for both steps 1 and 2, thus having Model B). We will prove below that, as the number of variables grows, Π almost certainly has no solution. Before presenting the proof, we note that according to the parameterization there are $p_2 D^2$ incompatible pairs in each constraint and, therefore, our condition is equivalent to $p_2 \geq 1/D$. Hence, our result implies that the currently used models are asymptotically uninteresting except, perhaps, for a small region of their parameter space (when $p_2 < 1/D$). We discuss this point further in section 3.1.

For every constraint R the probability, p_f, that some vertex (value) v in the domain of one of the two variables bound by R, is adjacent to (incompatible with) all the values in the domain of the other variable depends only on D. Moreover, since there are at least D edges in each constraint, this probability is a strictly positive constant (since D is a constant). We call such a value v

flawed. It is clear that a solution of Π cannot contain flawed values, since at least one constraint would be violated. Hence, if all the values in the domain of some variable are flawed, Π has no solution. For a variable bound by D or more constraints there is a strictly positive probability that all the values in its domain are flawed. Since in a random graph on n vertices (the constraint graph of an instance Π) with $\Theta(n)$ edges (constraints) there are, asymptotically, unboundedly many ($\Omega(n)$) vertices with degree a given constant, the probability that at least one variable will contain only flawed values gets, asymptotically, arbitrarily close to one. Therefore, almost certainly a random instance of CSP generated according to the currently used models is overconstrained. We make this argument formal below.

Consider the constraint graph G (of an instance Π). Recall, that the degree of a vertex in G is the number of constraints by which the corresponding variable is bound. The degree of each vertex obeys the binomial distribution of order n and with success probability the probability that a specific edge is selected. This probability is $p_1 = m_1/\binom{n}{2}$, where $m_1 = \Theta(n) = c \cdot n$. Therefore, $p_1 = 2c/n$. Since $p_1 \cdot n = 2c$, a constant and $p_1 \to 0$ as n tends to infinity, according to the Poisson limit theorem it holds that the degree of each vertex obeys the Poisson distribution with mean value $2c$, in the limit. (This conclusion holds for both options of Step 1.) Since the probability that a vertex has degree l is a constant p_l, it is concluded that, in the limit, for any constant l there almost certainly exist $p_l \cdot n = \Omega(n)$ vertices that have degree exactly l.

Consider now the following procedure: initialize S to contain all vertices of G that have degree exactly D. While S is not empty, pick a variable (vertex) v from S and (i) examine whether all values in its domain are flawed, (ii) remove from S the vertex v and all vertices adjacent to it in G.

Since initially $S = \Omega(n)$ and in each step we remove no more than $D + 1$ variables from S, the above procedure will be repeated $\Omega(n)$ times. Moreover, each time we examine a variable v the outcome is independent of all previous information "exposed" by the procedure, since none of the constraints that bind v was examined in previous steps (otherwise, v would have already been removed from S). Finally, the probability that all values in the domain of v are flawed is no less than the probability that each one of the D constraints binding v makes a distinct value in its domain flawed. This last probability is, clearly, p_f^D i.e., it depends only on D. Hence, the examination steps of the above procedure form a sequence of Bernoulli trials of length $\Omega(n)$ where the probability of success in each trial is lower bounded by a positive constant (i.e., independent of n). Thus, as n tends to infinity the probability that there exists a variable with all the values in its domain flawed, tends to one.

Alternatively, if we use the first option for Step 2 and take p_2 so that the expected number of incompatible pairs is D or more, then each constraint has a *constant* positive probability of having at least D incompatible pairs and the same line of reasoning as above may be followed.

Let us now assume that each constraint edge is chosen independently of the others and with probability $p_1 = \Theta(n)$ while each incompatibility edge, within a

constraint edge, is chosen independently of the others with probability p_2 (Model A). In this case, for some constant c, the expected number of constrained pairs of variables is cn. For each of these pairs of variables, and each pair of value assignments to them, the probability that this assignment is dissallowed is a positive constant (p_2). The probability, now, that all possible value assignments are disallowed, is also a positive constant and equal to $p_2^{D^2}$. Therefore, since there are $\Theta(n)$ constraint edges, the probability that for at least one of them all value assignments are disallowed, tends to 1 as n tends to infinity. Testing each constraint edge against dissallowance of any value assignment for the two variables it binds, may be seen as a sequence of (independent) Bernoulli trials with constant probability of success. Since the probability of success (i.e. dissallowance of any value assignment for a pair of variables) is a constant, the probability of failure in *all* $\Theta(n)$ trials, tends to 0 as n tends to infinity.

We wish to stress the fact that the appearance of flawed values is not the only deficiency that the currently used models may exhibit. As the Bernoulli trials argument shows, any local "flaw" which occurs with a constant, arbitrarily small but *positive* probability and which obstructs a variable from assuming a valid value, will occur with certainty, as the number of variables tends to infinity. Therefore, in the limit, all the instances will contain at least one variable to which no value can be assigned without violating some constraint.

3.1 An alternative model for random CSP instances

Before continuing with the definition of an alternative model which does not suffer from problems caused by local deficiencies, it should be emphasized that attempting to fix the old models simply by conditioning on each value having degree less than D in C, will probably not lead to any interesting new models. This is because, as we have already remarked, the old models generate many other *local* flaws similar to the one in the previous section. The one we point out is perhaps the simplest to describe (but not the most probable one). As an example, a pair of variables may be such that although it is possible to assign a value to each of them *separately*, it is impossible to assign compatible values to *both* of them simoultaneously (again for some local reasons as for the previous problem).

Moreover, in relation to the fact that the specific flaw we point out does not appear for $p_2 < 1/D$, our feeling is that $1/D$ is probably not a "critical" value for p_2 in the sense that "if $p_2 < 1/D$ then no trivial local inconsistencies arise". In fact, as we saw, in the case were the constraint edges are chosen with option (i) in Step 2, the models remain asympotically problematic for any constant $p_2 > 0$. Therefore, we believe that what is perhaps more important, and certainly more practically relevant, is a shift from constraints that contain an entirely random subset of $p_2 D^2$ forbidden pairs (even for $p_2 < 1/D$) to constraints where this subset has some structure. For example, assume that $p_2 = 1/D$ and that in each constraint between two variables v, w the i-th vertex (value) in the domain of v is incompatible with the j-th vertex in the domain of w iff $i = j$. Then, a random instance of this CSP encodes the problem of coloring a random graph with $p_1 n$

edges using D colors. This is an extremely interesting problem asymptotically, with numerous papers written on the subject (for an introduction see [2, 4]).

We feel that suggesting and analyzing interesting models that lack trivial local inconsistencies is a very worthwhile goal. In what follows, we put forward a model that is simple to state and analyze, as is the case with currently used models, which though does *not* suffer from trivial local inconsistencies. In this model, for a random CSP instance Π, it will be more intuitive to describe directly how to generate C^{Π} (as opposed to the two-step procedure described in Subsection 2.1).

Definition 3 (Model E). C^{Π} is a random n-partite graph with D vertices in each part constructed by uniformly, independently and with repetitions selecting $m = p\binom{n}{k}D^k$ hyperedges out of the $\binom{n}{k}D^k$ possible ones. Also, let $r = m/n$ denote the ratio of the selected edges to the number of variables.

Whenever we want to mention explicitly the parameters of the model, we will talk about "Model $\mathrm{E}_{n,m,D,k}$". We should mention that the constraints generated by Model E are similar to the *nogoods* used by Williams and Hogg ([31]), except that repetitions of edges (i.e. constraints) are allowed in order to simplify the analysis.

For binary constraint networks ($k = 2$) Model E, similarly to the currently used models, will only be interesting when the total number of constraints, m, is $\Theta(n)$. In this case, the expected number of repeated edges is rather insignificant ($O(1)$) and allowing repetitions in the definition simplifies the analysis of the model. Observe that if in each part of the n-partite graph described above we compress all D vertices into a single vertex, without eliminating any multiple occurrences of edges, we obtain a multigraph on n vertices with m edges. We denote this multigraph by G_m^{Π} (one can think of G_m^{Π} as the "constraint" multigraph). Since for instances generated according to Model E the m edges of the n-partite graph are selected uniformly, independently and with repetitions, it is easy to see that the same is true for the m edges of G_m^{Π} and hence G_m^{Π} is a random multigraph (in the $G_{n,m}$ model). The following two facts will be very useful:

Fact 4. *If for an instance Π of CSP, every connected component of G_m^{Π} has at most one cycle then Π has at least one solution.*

Fact 4 is well known for $D = 2$ (see [9] or [16], where a proof is given in terms of 2-SAT) and having $D > 2$ only makes things easier for that proof. Moreover:

Fact 5. *If a random multigraph G is formed by selecting uniformly, independently and with repetitions $\frac{n}{2}$ edges then, almost certainly, all the components of G will contain* at most *one cycle.*

Fact 5 is the analog of Theorem 5e in [12] when edges are selected with repetitions. ¿From facts 4 and 5, we get the following:

Theorem 6. *If for a random instance Π generated using Model E, C^{Π} has fewer than $\frac{n}{2}$ edges then Π almost certainly has a solution. Equivalently, if $r < 1/2$, then Π almost certainly has a solution.*

We believe that substantially stronger bounds can be derived by a more sophisticated analysis of this model and we leave this as future work. For constraint arities greater than 2, we present an analysis in the next section.

¿From the above discussion, we conclude that the instances generated by Model E avoid the trivial arymptotic insolubility that besets the instances generated by the currently used models. In the sections to follow, we will also provide a condition for asymptotic insolubility. When this condition is combined with Theorem 6 and Theorem 8 in the next section, a range is defined for the ratio r, outside of which we have certain solubility or certain insolubility. Therefore, it is conceivable that there exists a value r_0 within this range, for which the passage from solubility to insolubility is abrupt and, thus, displays a threshold-like behavior. This is strongly supported by Lemma 7 that is also presented in the next section.

We will additionally show how to obtain *improved* upper bounds for the probability that a random CSP instance generated according to Model E has a solution, as a function of the ratio r. We first use the well-known First Order Moment method in order to establish one such bound by counting the number of solutions of random CSP instances and we then present the Local Maxima method which, in essence, performs a more selective such counting.

4 Bounding the underconstrained region

In this section, we will show that for each $k \geq 3$, where k is the arity of the constraints, there exists a region for the value of $r = m/n$ such that, asymptotically, all the generated instences are underconstrained, i.e. they almost certainly have a solution. We will do so, using a result of Friedgut in [14] and the analysis of the *Unit Clause* (or UC for short) heuristic for k-SAT formulas of Chao and Franco ([7]). Since the adaptation does not present any difficulties, the main point of this section is to stress the fact that Model E avoids the main deficiency of the currently used models, namely that for any k and *any* r, however small, the generated instances are, asymptotically, almost certainly unsatisfiable.

Let $f_k(n, r)$ denote the probability that a random CSP instance generated according to Model $\mathrm{E}_{n,m,D,k}$ has a solution (remember that $r = m/n$). The following Lemma follows directly from the proof of the existence of a sharp threshold in random k-SAT, for any k, given in [14]:

Lemma 7. *For any k, there exists a function $r_k(n)$ such that for any $\epsilon > 0$*

$$\lim_{n\to\infty} f_k(n, r_k(n) - \epsilon) = 1 \text{ and } \lim_{n\to\infty} f_k(n, r_k(n) + \epsilon) = 0.$$

Therefore, for all $r \geq 0$ and $\epsilon > 0$,

$$\text{if } \lim_{n\to\infty} f_k(n, r) \geq \epsilon \text{ then } \lim_{n\to\infty} f_k(n, r) = 1.$$

The proof of this lemma follows verbatim the proof given in [14]. This lemma, essentially permits us to prove that random CSP instances fenerated according to Model $\mathrm{E}_{n,m,D,k}$ almost certainly have a solution, as the number of variables tends to infinity, by proving the weaker statement that they have a solution with probability $\epsilon > 0$, for some ϵ independent of n.

We will now present UC ([7]) adapted to CSP instances, which attempts to assign values to all variables, one at a time, using a simple heuristic rule: priority is always given to satisfy constraints of arity 1 We will provide a condition on $r = m/n$, which enables UC to find a solution, if it exists, to a random CSP instance, with probability greater than some $\epsilon > 0$, as the number of variables tends to infinity. In what follows, by C_1 we will denote the set of contraints of arity i.

The algorithm is the following:

Algorithm: Unit Clause (UC)
Input: An incombatibility hypergraph C
Output: A value assignment to all variables
1. **begin**
2. $t = 0$
3. **while** $t < n$
4. **if** $|C_1| > 0$ **then** choose, at random, a constraint $\{l_j\}$ from C_1 and
5. assign to variable j a value l'_j in $D_j - \{l_j\}$ at random
6. **else** choose, at random, a variable j not set yet
7. and assign it a value $l'_j \in D_j$ at random
8. remove all the constraints that contain a value in $D_j - \{l'_j\}$
9. remove the value l'_j from the constraints that contain it
10. $t \leftarrow t + 1$
11. **end**
12. **end**

As long as $|C_1| = 0$, the algorithm assigns, at each step, a value to a variable such that no constraint is violated. Now, if $|C_1| \neq 0$, there is a possibility that the attempt to satisfy a constraint from C_1 will produce at Step 9 an *empty* constraint, that is a constraint which was previously of arity 1 and which is not satisfied by the chosen value assignment.

Chao and Franco analyzed the behaviour of UC for k-SAT, as a function of the clauses to variables ratio r. ¿From their analysis, it is not hard to see that the following holds:

Theorem 8. *For random CSP instances generated according to Model E with constraints of arity $k \geq 3$ and domains of size D, the algorithm UC finds a solution with probability greater than ϵ, for some $\epsilon > 0$, if*

$$r = \frac{m}{n} < \frac{D^{k-1}}{k} \left(\frac{k-1}{k-2} \right)^{k-2}.$$

Theorems 6 and 8 establish, for any constraint arity $k \geq 2$, a region where random CSP instances generated according to Model E have, almost certainly, a solution. In the next section we will complement our study of Model E by establishing a region where the instances have, almost certainly, no solution.

5 Bounding the overconstrained region: the First Order Moment method

Let Π be a random CSP with n variables, generated in any of the ways defined so far. By $\mathcal{A}_n$ we denote the set of all value assignments for Π and by $\mathcal{S}_n$ we denote the random set of solutions of Π. We are interested in establishing a condition subject to which the probability that Π has a solution goes to 0 as $n \to \infty$. Such a condition is readily provided by the *First Order Moment* method (for excellent expositions see [2] and [27]). That is, by first noting that

$$E[|\mathcal{S}_n|] = \sum_{\Pi} (\Pr[\Pi] \cdot |\mathcal{S}_n(\Pi)|) \tag{1}$$

and then noting that

$$\Pr[\Pi \text{ has a solution}] = \sum_{\Pi} (\Pr[\Pi] \cdot I_{\Pi}), \tag{2}$$

where for an instantiation of the random variable Π the indicator variable I_{Π} is defined as

$$I_{\Pi} = \begin{cases} 1 \text{ if } \Pi \text{ has a solution,} \\ 0 \text{ otherwise.} \end{cases}$$

Hence, from (1) and (2) we get

$$\Pr[\Pi \text{ has a solution}] \leq E[|\mathcal{S}_n|]. \tag{3}$$

Calculating $E[|\mathcal{S}_n|]$ is much easier than calculating $\Pr[\Pi$ has a solution$]$. As an illustration we apply the First Order Moment method to the model we suggest, with $m = rn$. There are D^n possible value assignments and each one of the $\binom{n}{2}D^2$ possible incompatible pairs has a probability of $1/D^2$ to appear in a random value assignment. Since, we have m constraints,

$$\Pr[\Pi \text{ has a solution}] \leq E[|\mathcal{S}_n|] = D^n \left(1 - \frac{1}{D^2}\right)^m = \left(D\left(1 - \frac{1}{D^2}\right)^r\right)^n. \tag{4}$$

Hence, if $r > \ln(1/D)/\ln(1 - 1/D^2) \approx D^2 \ln D$ then $D(1 - \frac{1}{D^2})^r < 1$ and the probability that Π has a solution drops exponentially with n, asymptotically tending to zero.

6 Bounding the overconstrained region: the method of Local Maxima

The price paid for the simplicity of the First Order Moment method is that instances with a very large number of solutions, although they may occur with very small probability, contribute substantially to $E[|\mathcal{S}_n|]$. Hence, by substituting $|\mathcal{S}_n(\Pi)|$ for I_{Π} we might be "giving away" a lot. It is clear that if we could

manage to replace the set $|\mathcal{S}_n(\Pi)|$ by a smaller set, we would be deriving a quantity closer to $\Pr[\Pi \text{ has a solution}]$ and hence a tighter bound.

The technique introduced in [21], when applied to random CSP instances, takes advantage of the above observation and amounts to "compressing" $\mathcal{S}_n$ by requiring value assignments not only to be solutions of Π but to also satisfy a certain "local maximality" condition. The underlying intuition is that for a random solution of Π, if we choose a variable at random and change its value, the probability that we will end up with another valid solution is at least a constant $\rho > 0$. Consequently, solutions (when they exist) tend to appear in large "clusters" and instead of counting all solutions in a cluster we need only count a suitably defined representative one (locally maximum). We feel that this clustering is not specific to the model we put forward and that it is closely related to the notion of *influence* introduced in [18]. Hence, the technique could potentially apply to other models for random CSP. This technique has already been applied to random k-SAT [21] and random graph coloring [1], in each case giving the best bounds currently known.

Definition 9. For each variable in Π fix an arbitrary ordering of the values in its domain. The set $\mathcal{S}_n^\sharp$ is defined as the random set of value assignments A such that:

1. A is a solution of Π (written $A \models \Pi$), and
2. every value assignment obtained from A by changing the value of exactly one variable to some greater value, is not a solution of Π.

For a value assignment A, $A(X, v)$ will denote the value assignment obtained by changing the value assigned to variable X by A to a value v that is greater than the value assigned by A. The number of possible such changes for a value assignment A will be denoted by $sf(A)$.

Lemma 10. $\Pr[\Pi \text{ has a solution}] \leq E[|\mathcal{S}_n^\sharp|]$.

Proof. It is enough to observe that if Π has a solution then $\mathcal{S}_n^\sharp$ is not empty. The rest of the proof is identical to the proof of Equation (3). □

Since $\mathcal{S}_n$ can be written as the sum of D^n indicator variables, one for each possible value assignment A and nonzero iff A is a solution of Π, we obtain the following lemma.

Lemma 11. $E[|\mathcal{S}_n^\sharp|] = \Pr[A \in \mathcal{S}_n] \sum_{A \in \mathcal{A}_n} \Pr[A \in \mathcal{S}_n^\sharp \mid A \in \mathcal{S}_n]$.

6.1 The method of Local Maxima for Model E

In this section we apply the method of Local Maxima to random CSP instances generated using Model E, according to which we select uniformly a set of $m = rn$ incompatibility edges.

Theorem 12. *For a randomly generated binary CSP instance Π, let $\zeta = 1 - e^{-\frac{2r}{D^2-1}}$. If $(1-\frac{1}{D^2})^r \frac{1-\zeta^D}{1-\zeta} < 1$ then $\Pr[\Pi$ has a solution$]$ tends to 0 as n tends to infinity.*

Proof. As we argued earlier, $\Pr[A \in \mathcal{S}_n] = (1-\frac{1}{D^2})^m$. Thus, we only need to compute an upper bound of $\Pr[A \in \mathcal{S}_n^\sharp \mid A \in \mathcal{S}_n]$. Fix a value assignment $A \in \mathcal{S}_n$. Since $A \in \mathcal{S}_n$, every pair of values appearing in A is *not* adjacent (incompatible) in C. Consequently, conditioning on $A \in \mathcal{S}_n$ implies that the set of edges that might appear in Π has cardinality $D^2\binom{n}{2} - \binom{n}{2} = (D^2-1)\binom{n}{2}$. Now consider changing the value of variable X to a new value v greater than the value of X in A. The event $A(X,v) \notin \mathcal{S}_n$ occurs iff among the m edges in C, there is an edge connecting v with some other value in $A(X,v)$. Hence,

$$\begin{aligned}\Pr[A(X,v) \notin \mathcal{S}_n \mid A \in \mathcal{S}_n] &= 1 - \left(1 - \frac{n-1}{(D^2-1)\binom{n}{2}}\right)^m \\ &= 1 - \left(1 - \frac{2}{(D^2-1)n}\right)^{rn} \\ &= 1 - e^{-\frac{2r}{D^2-1}} + o(1).\end{aligned}$$

The events $A(X_i, v_j) \notin \mathcal{S}_n$, for each X_i, v_j are not independent. On the other hand, as we saw above, the set of constraint edges associated with each such event is disjoint with all other such sets. Intuitively, any such event $A(X_i, v_j) \not\models \Pi$ "exposes" only edges that can be of no harm to any other such event. Moreover, it exposes that at least one of the m edges was "used" to cause $A(X_i, v_j) \not\models \Pi$. Hence, the events $A(X_i, v_j) \not\models \Pi$ are in fact negatively correlated. Formally, this follows from the main Theorem in [23] by setting (i) $V = \{1, \ldots, m\}$, the set of selected constraints, (ii) $I = \{(i,j) | \text{for each } A(X_i, v_j)\}$, the index set of cardinality $sf(A)$, (iii) $X_v = (i,j)$ iff the vth selected constraint is not violated by A, but it is violated by $A(X_i, v_j)$, and (iv) for each i, $\mathcal{F}_i$ is the "increasing" collection of nonempty subsets of V. As a result we have,

$$\begin{aligned}\Pr[A \in \mathcal{S}_n^\sharp \mid A \in \mathcal{S}_n] &\leq \left(1 - \left(1 - \frac{2}{n(D^2-1)}\right)^{rn}\right)^{sf(A)} \\ &= \left(1 - e^{-\frac{2r}{D^2-1}} + o(1)\right)^{sf(A)} .\end{aligned}$$

If for each value assignment A we let k_i denote the number of variables assigned the i-th smallest value in their domain then, by Lemmas 10 and 11, we get:

$$\begin{aligned}&\Pr[\Pi \text{ has a solution}] \\ &\leq \left(1 - \frac{1}{D^2}\right)^m \sum_{A \in \mathcal{A}_n} \Pr[A \in \mathcal{S}_n^\sharp \mid A \in \mathcal{S}_n]\end{aligned}$$

$$\leq \left(1-\frac{1}{D^2}\right)^m \sum_{k_1,k_2,\ldots,k_D} \binom{n}{k_1,k_2,\ldots,k_D} \prod_{j=1}^{D} \left(1-e^{-\frac{2r}{D^2-1}}+o(1)\right)^{(D-j)\,k_j}$$

$$= \left(\left(1-\frac{1}{D^2}\right)^r \sum_{j=0}^{D-1}(1-e^{-\frac{2r}{D^2-1}})^j + o(1)\right)^n$$

$$= \left(\left(1-\frac{1}{D^2}\right)^r \frac{1-\zeta^D}{1-\zeta} + o(1)\right)^n, \text{ where } \zeta = 1-e^{-\frac{2r}{D^2-1}}.$$

□

Note, that the condition given by the theorem is more relaxed than the condition $\left(1-\frac{1}{D^2}\right)^r D < 1$, derived earlier by the First Order Moment method, since $\frac{1-\zeta^D}{1-\zeta} = \sum_{j=0}^{D-1}(1-e^{-\frac{2r}{D^2-1}})^j < D$. For example, when $D=3$ the First Order Moment method gives $r < 9.32$ while Theorem 12 gives $r < 8.21$.

When the arity of the constraints is $k > 2$, the following can be proved by extending in a straightforward manner the arguments given above:

Theorem 13. *For a randomly generated k-ary CSP instance Π according to Model E, if $(1-\frac{1}{D^k})^r \frac{1-\zeta^D}{1-\zeta} < 1$, where $\zeta = 1-e^{-\frac{kr}{D^k-1}}$, then $\Pr[\Pi$ has a solution$]$ tends to 0 as n tends to infinity.*

7 Conclusion

In this paper we attempted a first critical study of the asymptotic properties of some of the most commonly used models for generating random CSP instances. The main conclusion is that the currently *fixed constraint tightness* models are not suitable for the study of phase transition and threshold phenomena in random CSP because the instances they generate have, asymptotically, no solution. Considering Model B, for example, if the constraint tightness of a class of generated instances is $1/D^2$ or more, the instances are asymptotically overconstrained and, as a consequence, have no solution. The same is also true for Model A and, what is more, with no restrictions on the constraint tightness other than it must be greater than 0. Therefore, the picture of threshold-like behaviour that was exhibited by experiments conducted using these models is misleading.

It was shown, in particular, that a source (but not the only one) of this asymptotic insolubility is the appearance of "flawed" values, i.e. values which are incompatible with *all* the values of some other variable. However, it is rather clear that even if we modify the models in a way which explicitly avoids the generation of instances with flawed values, there will is still be no interesting asymptotic behavior because of other local faults. We believe that the main point of this conclusion is that the CSP community should focus on defining and analyzing new instance generation models in which the the set of forbidden pairs of values within a constraint possess some structure and is not completely random.

As a first step in this direction, we proposed in this paper an alternative model for generating random CSP instances, called Model E. This model resembles the model used for generating random boolean formulas for the satisfiability problem and the constraints it generates are similar to the "nogoods" proposed by Williams and Hogg ([31]). We showed that this model does not suffer from the deficiencies underlying the other popular models and it may indeed display an asymptotic threshold-like behavior, as we exhibited a range of its defining parameter (constraints to variables ratio) for which the generated instances have almost certainly a solution, as well as a different range for which they almost certainly have no solution. As we showed, this kind of behavior is not possible for the currently used models.

Moreover, we provided the Local Maxima method, which is an extension of the method used in [21] for the satisfiability problem, whereby it is possible to obtain a better bound for the overconstrained region than the bound obtained through the application of the First Order Moment method.

Acknowledgments

We would like to thank David Mitchell for his useful comments.

References

1. D. Achlioptas and M. Molloy, "Almost all graphs with 2.522 n edges are not 3-colorable," submitted.
2. N. Alon, J.H. Spencer, and P. Erdös, *The Probabilistic Method,* J. Wiley, New York (1992).
3. D.G. Bobrow and M. Brady, eds., *Special Volume on Frontiers in Problem Solving: Phase Transitions and Complexity*, Guest editors: T. Hogg, B.A. Hubermann, and C.P. Williams, *Artificial Intelligence*, Vol. 81, Numbers 1–2 (1996).
4. B. Bollobás, *Random Graphs*, Academic Press, London (1985).
5. A.Z. Broder, A.M. Frieze, and E. Upfal, "On the satisfiability and maximum satisfiability of random 3-CNF formulas," *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms* (1993) 322–330.
6. M.-T. Chao and J. Franco, "Probabilistic analysis of two heuristics for the 3-satisfiability problem," *SIAM Journal on Computing*, Vol. 15 (1986) 1106–1118.
7. M.-T. Chao and J. Franco, "Probabilistic analysis of a generalization of the unit-clause literal selection heuristic for the k-satisfiability problem," *Information Science,* Vol. 51 (1990) 289–314.
8. P. Cheeseman, B. Kanefsky and W. Taylor, "Where the really hard problems are," *Proceedings of IJCAI '91* (1991) 331–337.
9. V. Chvátal and B. Reed, "Mick gets some (the odds are on his side)," in *Proceedings of 33rd IEEE Symposium on Foundations of Computer Science* (1992) 620–627.
10. R. Dechter, "Constraint networks," in S. Shapiro (ed.), *Encyclopedia of Artificial Intelligence*, Wiley, New York, 2nd ed. (1992) 276–285.
11. O. Dubois and Y. Boufkhad, *A General Upper Bound for the Satisfiability Threshold of Random r-SAT Formulae,* Preprint, LAFORIA, CNRS-Université Paris 6, 1996.

12. P. Erdös and A. Rényi, "On the evolution of random graphs," *Publ. of the Math. Inst. of the Hung. Acad. Sci.,* Vol. 5 (1960) 17–61.
13. A. El Maftouhi and W. Fernandez de la Vega, "On random 3-SAT", *Combinatorics, Probability and Computing,* Vol. 4 (1995) 189–195.
14. E. Friedgut, "Sharp thresholds for graph properties and the k-sat problem," *Manuscript in preparation.*
15. A. Frieze and S. Suen, "Analysis of two simple heuristics on a random instance of k-SAT," *Journal of Algorithms*, Vol. 20 (1996) 312–355.
16. A. Goerdt, "A threshold for unsatisfiability," *Journal of Computer and System Sciences*, Vol. 53 (1996) 469–486.
17. T. Hogg, "Refining the phase transition in combinatorial search," in [3], 127–154.
18. J. Kahn, G. Kalai, and N. Linial, "The influence of variables on Boolean functions", *Proceedings of the 29th Annual Symposium on the Foundations of Computer Science* (1988) 68–80.
19. A. Kamath, R. Motwani, K. Palem, and P. Spirakis, "Tail bounds for occupancy and the satisfiability threshold conjecture," *Random Structures and Algorithms*, Vol. 7 (1995) 59–80.
20. S. Kirkpatrick and B. Selman, "Critical behavior in the satisfiability of random Boolean expressions," *Science* 264, pp 1297–1301, 1994.
21. L.M. Kirousis, E. Kranakis, D. Krizanc, and Y.C. Stamatiou, "Approximating the unsatisfiability threshold of random formulas," *Random Structures and Algorithms*, Vol. 12 (1998) 253–269.
22. A.K. Mackworth, "Constraint satisfaction," in S. Shapiro (ed.) *Encyclopedia of Artificial Intelligence*, Wiley, New York, 2nd ed. (1992) 285–293.
23. C. McDiarmid, "On a correlation inequality of Farr," *Combinatorics, Probability and Computing*, Vol. 1 (1992) 157–160.
24. D. Mitchell, B. Selman, and H. Levesque, *Generating hard satisfiability problems*, Artificial Intelligence, Vol. 81 (1996), 17–29.
25. P. Prosser, "An empirical study of phase transitions in binary constraint satisfaction problems," in [3], 81–109.
26. B.M. Smith and M.E. Dyer, "Locating the phase transition in binary constraint satisfaction problems," in [3], 155-181.
27. J. H. Spencer, *Ten Lectures on the Probabilistic Method*, 2nd edition, SIAM, Philadelphia (1994).
28. D. Waltz, "Understanding line drawings of scenes with shadows," *The Psychology of Computer Vision*, McGraw-Hill, New York (1975) 19–91.
29. C. Williams and T. Hogg, "Using deep structure to locate hard problems," *Proceedings of AAAI-92* (1992) 472–477.
30. C. Williams and T. Hogg, "Extending deep structure," *Proceedings of AAAI-93*, (1993) 152–158.
31. C. Williams and T. Hogg, "Exploiting the deep structure of constraint problems," *Artificial Intelligence*, Vol. 70 (1994) 73–117.

This article was processed using the LaTeX macro package with LLNCS style