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

(contact author: Yannis C. Stamatiou)

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" behavior. In this spirit, some preliminary theoretical work has been done in analyzing these models asymptotically, i.e., as the number of variables grows. 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, suffering 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 exhibits non-trivial asymptotic behavior. Moreover, for this new model 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 [12, 27, 33].

Following the seminal work of Cheeseman, Kanefsky and Taylor [10], 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 [5]. 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 to decide for each instance in the set whether a solution exists (using heuristic methods since, in general, CSP is NP-complete). The proportion of instances that have a solution is used as an indication of the "likelihood", while the median time taken, per instance, captures a notion of "hardness".

Research in this direction originated from research on random instances of "specific" CSP problems such as SAT and graph colorability (see [10]). Especially, for random instances of SAT, Mitchell, Selman and Levesque ([29]) 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. A "50%-solvability" 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 as connections of this threshold behavior to phase transition phenomena in physics are described in [24].

The experimental results have motivated a theoretical interest in understanding the asymptotic behavior of random k-SAT formulas. In the following, we give a brief summary of the strongest known such 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 [11, 19]: 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 [8, 9, 7, 15, 22, 17, 25, 13] narrowing the area for which we do not have almost certain (un)satisfiability. The best bounds currently known come from [17] where it was proven that a random instance of 3-SAT is almost certainly satisfiable if $r < 3.003$, and [25] 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 $\Theta(2^k/k)$ [9, 11, 17] while 2^k is an easy upper bound and was improved by a constant factor in [25] by extending the techniques used for 3-SAT.

Recently, Friedgut [16] made tremendous progress towards establishing the *existence* of a threshold for k-SAT for all values of k. Let m denote the number of clauses in a random k-SAT formula. Friedgut showed that for every $k \geq 2$, there exists a *function* $r_k(n)$ such that for m around $r_k(n)n$ the probability of satisfiability drops suddenly from near 1 to near 0. While it is widely believed that $\lim_{n\to\infty} r_k(n)$ exists, and thus equals r_k, this has not been established yet; moreover, it is important to note that Friedgut's approach does not provide any information regarding the value of r_k. More recently, using Friedgut's approach, it was shown in [1] that an analogous situation exists for the probability that a random graph on n vertices and $m = rn$ edges is k-colorable.

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 [10, 20, 30, 31, 34, 35, 36]. A first step towards an asymptotic analysis of Random CSP was made by Smith and Dyer in [31], who examined a well-established model for generating random CSP with binary constraints, Model B described in Section 2.1, along with other models. Using the *first moment* method they derived 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 studied the variance of the number of solutions as a function of the defining parameters.

In this paper we show that as long as the number of permissible pairs of value assignments in each constraint is any constant fraction of all possible pairs, then as the number of variables increases, the probability that a solution exists tends to 0. Obviously, this implies that a threshold phenomenon *does not* underlie Random CSP generated using such models, 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: when in each constraint, every pair of value assignments is forbidden *randomly and with constant probability*, each variable has *constant* positive probability of being "bad". That is, the variable cannot be set to any of the values in its domain so as to satisfy all the constraints binding it, even if all other variables are free to take any value. Clearly, a bad variable makes a CSP unsatisfiable, albeit in a trivial, completely local manner. Asymptotically, this constant positive probability of "badness" for a fixed variable will be shown to imply that almost certainly there exists at least one bad variable and thus that the instance is almost certainly unsatisfiable. It is worth noting that this phenomenon does not occur in "specific" random CSP (such as SAT or colorability), where there is some structure, or even no randomness at all, in the choice of the forbidden pairs of values for two constrained variables.

As a result, we conclude that new models for generating random CSP are necessary 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 it generates do 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 different bound. The analysis for the underconstrained region parallels the analysis for random k-SAT in [9], while the counting argument is based on a technique introduced in [25].

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 and 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.

We describe below the generation of random CSP instances with *binary* constraints, as this will be sufficient for our purposes while the generalization to instances with constraints of higher arity is completely straightforward. Fixing n and D, the generation of a random binary CSP instance 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 incompatibility graph). A general framework for doing this is presented in [30, 31]. 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*.

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 [31], 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.

It is not hard to see that for all four models above, if the (expected) number of constraints is cn, for some suitably large constant c, then the resulting instances are almost certainly overconstrained. This follows from a straightforward application of the first moment method as there are D^n possible assignments to the variables and each non-empty constraint is satisfied with probability at most $(1 - 1/D^2)$. Hence, the expected number of solutions is bounded by $D^n(1 - 1/D^2)^{f(n)}$, where $f(n)$ is the number of non-empty constraints. As a result, for any $p_2 \neq 0$, if $p_1 = c/n$, for large enough $c = c(p_2, D)$, almost certainly $f(n) > (1 + \epsilon)\rho n$, where $D(1 - 1/D^2)^\rho = 1$. Hence, for such c, the expected number of solutions tends to 0 as $n \to \infty$ and, thus, so does the probability of satisfiability.

In the next section we show that, in fact, for *any* $p_1 = c/n$ all four models generate instances that are trivially unsatisfiable. On the other hand, if $p_1 = c/n^{1+\frac{1}{q}}$ then almost certainly the largest component of the constraint graph contains only $q + 1$ variables (see [6], p.90) making for trivial CSP instances.

3 Shortcomings of currently used models

Let us say that a value d in the domain of some variable v is "flawed" if there exists a constraint R in Π, binding v, such that d is adjacent to (incompatible with) all the values in the domain of the other variable in R. It is clear that solutions of Π cannot contain flawed values, since at least one constraint would be violated. Moreover, if all the values in the domain of some variable v are flawed, then v is a bad variable, and Π is unsatisfiable, since v cannot be set to any of the values in its domain, even if all other variables are free to take any value.

If d is a value in the domain of a variable bound by a constraint then, clearly, the probability, p_f, that d is flawed because of that particular constraint depends only on p_2, D. In particular, if Step 2 is done in fashion (i) then $p_f > 0$ for *any* $p_2 \neq 0$; if Step 2 is done in fashion (ii) then $p_f > 0$ for all $p_2 \geq 1/D$, as there are D^2 potential incompatibility edges. We will prove below that if $p_f > 0$, then, as the number of variables grows, the random instances generated almost certainly have no solution. As a result, we conclude that the currently used models for random CSP are asymptotically uninteresting except, perhaps, for a small region of their parameter space (when $p_2 < 1/D$ and Step 2 is done in fashion (ii)). We will discuss this last point further in the end of this section 3.1.

To see why having $p_f > 0$ implies almost certain insolubility, first note that the choice of incompatible pairs in each constraint is completely independent from the analogous changes in other constraints. Hence, since D is constant, if a variable is bound by D or more constraints there is a strictly positive constant probability that all the values in its domain become flawed. Now, since in a random graph on n vertices (the constraint graph of an instance Π) with $\Theta(n)$ edges (constraints) there are 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 1. 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. If Step 1 is performed **in** fashion (i) then clearly, the degree of each vertex obeys the binomial distribution with $n-1$ trials and probability of success $p_1 = c/n$. Since the expected degree is bounded, it is well-known that as $n \to \infty$ the degree of a given vertex is distributed as a Poisson random variable with mean c. In fact, something stronger is true: almost certainly, for each $i \geq 0$, the number of vertices of degree i is $\mathrm{Po}(c;i) \cdot n + o(n)$, where $\mathrm{Po}(c;i)$ is the probability that a Poisson distributed random variable with mean c takes the value i. Moreover, this also holds if Step 1 is performed in fashion (ii). (Proofs for both cases can be found in [6].) Hence, in the following we can assume that, independently of how Step 1 is performed, there are $\Omega(n)$ vertices of degree D.

Consider 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. Since the choice of incompatible pairs in each constraint is completely independent from the analogous changes in other constraints, this probability is at least $p_f^D > 0$. 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 $n \to \infty$, the probability that there exists a variable with all the values in its domain flawed tends to 1.

Note that if Step 2 is done in fashion (ii) we do not even need to consider flawed values since each constraint has probability $p_2^{D^2}$ of forbidding *all* D^2 value-combinations of the bound variables. Hence, in that context we could derive asymptotic insolubility by simply noting that we have $\Omega(n)$ trials, as many as the constraints, each with constant probability of success.

Remark: As a summary of the arguments presented above, we expand below on the role of the parameter p_2 in the appearance of the trivial local inconsistencies we described (we will assume that the number of edges in the constraint graph is, almost certainly, $\Theta(n)$).

If Step 2 is done in fashion (i) then p_2 is irrelevant since, as we showed above, for *any* $p_2 \neq 0$ the instances generated are almost certainly unsatisfiable. However, if Step 2 is done in fashion (ii) we could prove the appearance of trivial inconsistencies (caused by "bad" variables), only for $p_2 \geq 1/D$ as *only then* "flawed" values occur. In fact, Model B is known to generate problems that *do have* a threshold behavior for certain $p_2 < 1/D$. One such example is random 2-SAT, which corresponds to binary CSP instances generated using Model B, with $D = 2$ and $p_2 = 1/4$. As we mentioned in Section 1, it is well-known that 2-SAT has a sharp threshold occurring around $p_1 = 2/n$: for any $\epsilon > 0$, almost all formulas with $m = (1 - \epsilon)n$ clauses are satisfiable, while almost all formulas with $m = (1 + \epsilon)n$ clauses are unsatisfiable. Moreover, Friedgut [16] proved the existence of sharp threshold behavior (albeit in a non-uniform sense) for random k-SAT, for all $k \geq 2$, i.e., for Model B where $D = 2$ and $p_2 = 1/D^2$.

In fact, since the appearance of [2], Gent [18] and, independently, Ke [23] have shown that for $p_2 < 1/D$, if Step 2 is performed in fashion (ii) then there exists a range of $p_1 = c/n$ in which the resulting instances do not suffer from trivial asymptotic insolubility.

We believe that rather than focusing further on the region $p_2 < 1/D$, it is perhaps more important, and certainly more practically relevant, to shift from constraints that contain a random subset of $p_2 D^2$ forbidden pairs of values, to constraints where this subset has some structure. For example, consider CSP where $p_2 = 1/D$ and in each constraint between variables v, w, the i-th vertex (value) in the domain of v is incompatible precisely with the i-th vertex in the domain of w. Then, a random instance of such CSP encodes the problem of coloring a random graph with $p_1\binom{n}{2}$ edges using D colors. This is an extremely interesting problem asymptotically, exhibiting a sharp threshold behavior [1] and appearing computationally difficult around its threshold [10].

Suggesting and analyzing interesting models that lack trivial local inconsistencies appears to be 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 indeed does *not* suffer from trivial local inconsistencies. We want to point out that we view this model only as a first step, intended to indicate the possibility of non-trivial asymptotic behavior for random CSP, and we hope that developing more sophisticated, and perhaps more realistic, models will be of interest to the CSP community. In particular, we feel that models which impose structure in the choice of forbidden edges while maintaining amenability to asymptotic analysis would be most welcome.

3.1 An alternative model for random CSP instances

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 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 ([36]), except that here repetitions of edges (i.e., constraints) are allowed, simplifying the analysis greatly.

For binary constraint networks ($k = 2$) Model E, similarly to the currently used models, is only 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 [11] or [19], where a proof is given in terms of 2-SAT) and having $D > 2$ only makes things easier for that proof. Moreover:

Fact 5. *For any $\epsilon > 0$, if a random multigraph G is formed by selecting uniformly, independently and with repetitions $(\frac{1}{2} - \epsilon)n$ edges then, almost certainly, all the components of G will contain* at most *one cycle.*

Fact 5 is the analog of Theorem 5e in [14] when edges are selected with repetitions. Facts 4 and 5, imply:

Theorem 6. *If for a random instance Π generated using Model E, we have $r < 1/2$ then, since C^{Π} has rn edges, Π almost certainly has a solution.*

In the next section we will derive a stronger lower bound for r which does not rely on the underlying constraint graph having trivial structure. That lower bound will follow by extending the analysis of Chao and Franco for random k-SAT [8] to Model E with constraints of arbitrary arity $k \geq 2$. We believe that even stronger bounds can be derived by a more sophisticated analysis of this model and we leave this as future work.

Theorem 6 allows us to conclude that the instances generated by Model E avoid the trivial asymptotic insolubility that besets the instances generated by the currently used models. In the sections to follow, we will also provide successively tighter conditions for asymptotic insolubility. When those are combined with Theorem 6, and Theorem 8 of the next section, a range is defined for the ratio r, outside of which we have almost certain solvability or almost certain insolubility. Therefore, it is conceivable that there exists a value r_0 within this range, for which the passage from solvability 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.

4 Bounding the underconstrained region

In this section we show that for each $k \geq 2$, where k is the arity of the constraints, there exists r_k such that if $m/n < r_k$ then the instances generated almost certainly have a solution. We do so, by considering a natural extension of the *Unit Clause* heuristic for k-SAT of Chao and Franco [9] (UC for short) and applying a result of Friedgut [16]. the

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 $k \geq 2$, given in [16]:

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 $\epsilon > 0$,

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

The proof of this lemma follows verbatim the proof given in [16]. This lemma, permits us to prove that random CSP instances generated according to Model $\mathrm{E}(n, m, D, k)$ almost certainly have a solution 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 ([9]) 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 $r = m/n$, under which the probability that UC finds a solution to a random CSP instance, is greater than some $\epsilon > 0$, independent of n. In what follows, C_1 will denote the set of constraints of arity 1.

The algorithm is the following:

Algorithm: Unit Clause (UC)
Input: An incompatibility hypergraph C
Output: A value assignment to all variables
1. **begin**
2. $\quad t = 0$
3. $\quad$ **while** $t < n$
4. $\qquad$ **if** $|C_1| > 0$ **then** choose, at random, a constraint $\{l_j\}$ from C_1 and
5. $\qquad\qquad$ assign to variable j a value l'_j in $D_j - \{l_j\}$ at random
6. $\qquad\quad$ **else** choose, at random, a variable j not set yet
7. $\qquad\qquad$ and assign it a value $l'_j \in D_j$ at random
8. $\qquad$ remove all the constraints that contain a value in $D_j - \{l'_j\}$
9. $\qquad$ remove the value l'_j from the constraints that contain it
10. $\qquad t \leftarrow t + 1$
11. $\quad$ **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 behavior of UC for k-SAT, as a function of the clauses to variables ratio r. Following their analysis, it is straightforward to see that the following holds:

Theorem 8. *For random CSP instances generated according to Model E with constraints of arity $k \geq 2$ and domains of size D, if r is such that for all $x \in [0, 1]$,*

$$\frac{1}{D^{k-2}} \binom{k}{2} (1 - x) x^{k-2} r < \frac{D}{2},$$

then UC finds a solution with probability greater than ϵ, for some $\epsilon = \epsilon(r) > 0$. For $k = 2$ this condition amounts to $r < \frac{D}{2}$ while for $k > 2$ to $r < \frac{D^{k-1}}{k} \left(\frac{k-1}{k-2}\right)^{k-2}$.

The intuition behind the proof of this theorem, as in the proof given for UC in [9], is as follows: after the algorithm has successfully assigned values to the first t variables, there are $S_i(t)$ constraints of arity i that are uniformly distributed among all possible constraints of length i on the unset variables. Moreover, as long as the expected number of constraints of arity 1 generated per step remains below 1, such constraints do not "accumulate" (analogously to a queue in a stable server). As a result, the probability that in any given step there will appear two constraints with arity 1 requiring a variable to admit different values is very small (recall that this is the only scenario under which the algorithm fails). The condition presented in Theorem 8 amounts precisely to preserving the aforementioned expectation below 1 throughout the execution of the algorithm.

Theorem 8 establishes, 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 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 moment* method (for excellent expositions see [4] and [32]). That is, by first noting that

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

and then noting that

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

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 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,

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

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 0.

6 Bounding the overconstrained region: the method of Local Maxima

The price paid for the simplicity of the first 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 $\Pr[\Pi$ has a solution$]$ and hence a tighter bound.

The technique introduced in [25], 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 $\rho = \rho(k, r) > 0$. 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 [21]. Hence, the technique could potentially apply to other models for random CSP. This technique has already been applied to random k-SAT [25] and random graph coloring [3], in each case giving the best bounds currently known.

Before continuing, we would like to point out that in a previous version of our paper ([2]), Section 6 contained an error in the analysis of Model B. Since the correct analysis requires abandoning a number of simplifying assumptions and the improvement obtained over the first moment method is rather negligible, we have removed that section completely in the present paper.

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 $n \to \infty$.*

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,

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

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 [28] 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,

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

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:

$$\begin{aligned}&\Pr[\Pi \text{ has a solution}] \\ &\leq \left(1 - \frac{1}{D^2}\right)^m \sum_{A \in \mathcal{A}_n} \Pr[A \in \mathcal{S}_n^\sharp \mid A \in \mathcal{S}_n] \\ &\leq \left(1 - \frac{1}{D^2}\right)^m \sum_{k_1, k_2, \ldots, k_D} \binom{n}{k_1, k_2, \ldots, k_D} \prod_{j=1}^{D} \left(1 - e^{-\frac{2r}{D^2-1}} + O(1/n)\right)^{(D-j)\,k_j} \\ &= \left(\left(1 - \frac{1}{D^2}\right)^r \sum_{j=0}^{D-1} (1 - e^{-\frac{2r}{D^2-1}})^j\right)^n \times O(1) \\ &= \left(\left(1 - \frac{1}{D^2}\right)^r \frac{1 - \zeta^D}{1 - \zeta}\right)^n \times O(1), \text{ where } \zeta = 1 - e^{-\frac{2r}{D^2-1}}.\end{aligned}$$

□

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 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 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 $n \to \infty$.*

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 used models are not suitable for the study of phase transition and threshold phenomena because the instances they generate asymptotically have no solutions almost certainly. In particular, Model A generates almost certainly unsatisfiable instances for every $p_2 \neq 0$, while Model B generates almost certainly unsatisfiable instances for every $p_2 \geq 1/D$ (analogously for the other two models).

We showed that one source of this asymptotic insolubility is the appearance of "flawed" values, i.e., values which are incompatible with *all* the values of some other variable. As reported in [26], a number of experimental studies have avoided this pitfall but many others did not.

We believe that the main point of our work lies in demonstrating the need for defining and analyzing generation models for random CSP which avoid trivial inconsistencies. In particular, our results suggest that in the traditional CSP models, the set of forbidden pairs of values in each constraint should possess some structure and not be completely random (at least for most of the "constraint tightness" range.)

As a first step in defining new random CSP models, we proposed 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 ([36]). 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 always possible for the currently used models.

Moreover, we provided the Local Maxima method, which is an extension of the method used in [25] 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 moment method.

Acknowledgments

We would like to thank David Mitchell for useful discussions and the anonymous referees whose comments greatly improved the presentation of the paper.

References

1. D. Achlioptas and E. Friedgut, "A sharp threshold for k-colorability," Random Structures and Algorithms, Vol. 14 (1999) 63-70.
2. 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. Third International Conference on Principles and Practice of Constraint Programming (CP 97)*, Springer-Verlag, pages 107–120, 1997.

3. D. Achlioptas and M. Molloy, “Almost all graphs with 2.522 n edges are not 3-colorable,” Electronic Journal of Combinatorics, to appear.
4. N. Alon, J.H. Spencer, and P. Erdös, *The Probabilistic Method,* J. Wiley, New York (1992).
5. 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).
6. B. Bollobás, *Random Graphs*, Academic Press, London (1985).
7. 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.
8. 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.
9. 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.
10. P. Cheeseman, B. Kanefsky and W. Taylor, “Where the really hard problems are,” *Proceedings of IJCAI '91* (1991) 331–337.
11. 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.
12. R. Dechter, “Constraint networks,” in S. Shapiro (ed.), *Encyclopedia of Artificial Intelligence*, Wiley, New York, 2nd ed. (1992) 276–285.
13. 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.
14. 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.
15. A. El Maftouhi and W. Fernandez de la Vega, “On random 3-SAT”, *Combinatorics, Probability and Computing,* Vol. 4 (1995) 189–195.
16. E. Friedgut, “Sharp thresholds for graph properties and the k-sat problem,” *Journal of the American Matematical Society*, to appear.
17. 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.
18. I. Gent, personal communication, 1999.
19. A. Goerdt, “A threshold for unsatisfiability,” *Journal of Computer and System Sciences*, Vol. 53 (1996) 469–486.
20. T. Hogg, “Refining the phase transition in combinatorial search,” in [5], 127–154.
21. 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.
22. 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.
23. X. Ke, personal communication, 1999.
24. S. Kirkpatrick and B. Selman, “Critical behavior in the satisfiability of random Boolean expressions,” *Science* 264, pp 1297–1301, 1994.
25. 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.

26. E. MacIntyre, Patrick Prosser, Barbara Smith, and Toby Walsh, "Random Constraint Satisfaction: theory meets practice," in Proc. 4th International Conference on Principles and Practice of Constraint Programming (CP' 98), 1998.
27. A.K. Mackworth, "Constraint satisfaction," in S. Shapiro (ed.) *Encyclopedia of Artificial Intelligence*, Wiley, New York, 2nd ed. (1992) 285–293.
28. C. McDiarmid, "On a correlation inequality of Farr," *Combinatorics, Probability and Computing*, Vol. 1 (1992) 157–160.
29. D. Mitchell, B. Selman, and H. Levesque, *Generating hard satisfiability problems*, Artificial Intelligence, Vol. 81 (1996), 17–29.
30. P. Prosser, "An empirical study of phase transitions in binary constraint satisfaction problems," in [5], 81–109.
31. B.M. Smith and M.E. Dyer, "Locating the phase transition in binary constraint satisfaction problems," in [5], 155-181.
32. J. H. Spencer, *Ten Lectures on the Probabilistic Method*, 2nd edition, SIAM, Philadelphia (1994).
33. D. Waltz, "Understanding line drawings of scenes with shadows," *The Psychology of Computer Vision*, McGraw-Hill, New York (1975) 19–91.
34. C. Williams and T. Hogg, "Using deep structure to locate hard problems," *Proceedings of AAAI-92* (1992) 472–477.
35. C. Williams and T. Hogg, "Extending deep structure," *Proceedings of AAAI-93*, (1993) 152–158.
36. C. Williams and T. Hogg, "Exploiting the deep structure of constraint problems," *Artificial Intelligence*, Vol. 70 (1994) 73–117.