Random Constraint Satisfaction
A More Accurate Picture

Dimitris Achlioptas[*§]
(optas@cs.toronto.edu)

Lefteris M. Kirousis[†¶]
(kirousis@ceid.upatras.gr)

Evangelos Kranakis[‡‖]
(kranakis@scs.carleton.ca)

Danny Krizanc[‡‖]
(krizanc@scs.carleton.ca)

Michael S.O. Molloy[*‖]
(molloy@cs.toronto.edu)

Yannis C. Stamatiou[†¶]
(stamatiu@fryni.ceid.upatras.gr)

Abstract

Recently there has been a great amount of interest in Random Constraint Satisfaction Problems, both from an experimental and a theoretical point of view. Rather intriguingly, experimental results with various models for generating random CSP instances suggest a "threshold-like" behavior and some theoretical work has been done in analyzing these models when the number of variables becomes large (asymptotic). In this paper we prove that the models commonly used for generating random CSP instances *do not* have an asymptotic threshold. In particular, we prove that as the number of variables becomes large *almost all* instances they generate are trivially overconstrained. We then present a new model for random CSP and, in the spirit of random k-SAT, we derive lower and upper bounds for its parameters so that instances are "almost surely" underconstrained and overconstrained, respectively. Finally, for the case of one of the popular models in Artificial Intelligence we derive sharper estimates for the probability of being overconstrained as a function of the number of variables.

*Department of Computer Science, University of Toronto, Toronto, ON M5S 3G4, Canada.

†Department of Computer Engineering and Informatics, University of Patras, Rio, 265 00 Patras, Greece.

‡School of Computer Science, Carleton University, Ottawa, ON K1S 5B6, Canada.

§Supported by an NSERC (Natural Sciences and Engineering Research Council) PGS B Scholarship.

¶Partially supported by the EU ESPRIT Long-term Research Project ALCOM-IT (Project Nr. 20244)

‖Supported in part by an NSERC grant.

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 complete definition 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 applications ranging from scene labeling to scheduling and knowledge representation [10, 20, 25].

In recent years, 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 may be discovered. An extensive account of relevant results, both experimental and theoretical, can be found in [3]. One of the most commonly used practices 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 heuristics for deciding if a solution exists (in general, CSP is NP-complete). 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. A characteristic that is shared by many of these experiments is that the fraction of value assignments that is permitted for the variables bound by a constraint is kept *constant* as the number of variables increases.

For random instances of k-SAT, which is a special case of CSP, a very dramatic behavior has been observed. We will say that a sequence of events $\mathcal{E}_n$ occurs "almost surely" if $\Pr(\mathcal{E}_n)$ tends to one as n tends to infinity. Let r denote the ratio of clauses to variables in an instance of k-SAT. For $k = 2$ a sharp threshold was proved in [9, 15]: A random instance of 2-SAT is almost surely satisfiable if $r < 1$ and almost surely unsatisfiable if $r > 1$. For $k = 3$, such a sharp threshold behavior is widely conjectured to exist and there has been a series of papers [6, 7, 5, 12, 18, 14, 19] narrowing the area for which we do not have almost surely (un)satisfiability. The best bounds currently known come from [14] where it was proven that a random instance of 3-SAT is almost surely satisfiable if $r < 3.003$, and [19] where it was proven that a random instance of 3-SAT is almost surely unsatisfiable if $r > 4.598$. For general k, the best known lower bound for r is $\Theta(2^k/k)$ [7, 9, 14] while 2^k is an easy upper bound and was improved by a constant factor in [19] by extending the techniques used for 3-SAT. Very recently, Friedgut [13] made great progress towards establishing the *existence* of a threshold for k-SAT for all values of k, although he does not determine its asymptotic location.

There is accumulating *experimental* evidence that there exists a *critical region* for the defining parameters of random CSP. That is, determining the existence of a solution tends to be hard for randomly generated instances with defining parameters in this region. 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 tends to be easy. Moreover, the "hard" region seems to narrow as the number of variables increases. This has led to the belief that a threshold phenomenon underlies random CSP when the number of variables, n, becomes asymptotic [8, 16, 22, 23, 26, 27, 28] in a spirit similar to random $k-$SAT. In [23], Smith and Dyer examine a well-established model for generating random CSP with binary constraints. Using the first moment method they derive a condition on the defining parameters subject to which the probability that a random instance has a solution tends to zero as n tends to infinity. Thus they show that a random CSP with defining parameters in the region defined by this condition almost surely does not have a solution. 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 fraction of permissible pairs of value assignments in each constraint is a constant (as, e.g., it is assumed in [23]), then as the number of variables increases, the probability that a random CSP has a solution tends to zero. This implies that a threshold phenomenon *cannot* underlie random instances of CSP generated using the current models, including those examined in [23], since asymptotically *almost all* instances generated, will not have a solution. The threshold-like picture given by experimental results is misleading, since the problems with defining parameters in what is currently perceived as the underconstrained region (because a solution can be found fast) are in fact overconstrained for large n (obviously, larger than the values used in experiments). This indicates that a new parameterization (model) is necessary for random CSP before any asymptotic analysis can be performed.

We make a first step in this direction by proposing a simple new model and analyzing it. That is, we show how to almost surely find a solution for a random CSP if its defining parameters lie within an area of the parameter space of the new model (underconstrained problems) and we also provide a counting argument showing that no solution exists for random CSP with defining parameters in another area of the parameter space (overconstrained problems). The analysis for the underconstrained region exploits a connection to random graphs while the counting argument is based on a technique introduced by Kirousis, Kranakis and Krizanc in [19]. Finally, we analyze the popular model (Model B) examined in [23], and derive a sharper upper bound for the probability of having a solution than the one provided by the first moment method [23]. Since, as we show, this model is in fact unsuitable for asymptotic analysis, our bound is a function of the defining parameters *and* n and we hope that it might prove useful in setting up experiments.

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 if a solution exists.

For an instance Π of CSP with n variables, its *constraint hypergraph* G^{Π} (or just G when no confusion may arise) 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 a convenient graph-theoretic representation of a CSP instance Π, defined as follows: The *incompatibility hypergraph* of Π, C^{Π} (or just C, when no confusion may arise) 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. 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.

In order to keep the exposition that follows more intuitive and less technical, we will detail our arguments only for CSP with constraints of arity 2. The generalization to CSP with constraints of arity $k > 2$ is straightforward.

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 $D \geq 2$. Hence, fixing n and D, the generation of a random CSP is usually done in two steps. First, the constraint graph G 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. A general framework for doing this is presented in [22, 23]. More precisely,

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

Step 2 *Either for every edge of G each one of the D^2 edges in C is selected with probability p_2, or 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*. 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, p_1 is varied so that there are $\Theta(n)$ incompatible pairs of values in total. 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.

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 [23], if both 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 pertains to all four models.

3 Shortcomings of currently used models for random CSP

Let Π be a random CSP with D or more incompatible pairs in each constraint (suppose that we use the second option for Step 2). We will prove below that, as the number of variables grows, Π almost surely 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 so 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 surely 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. Since the total number of edges (constraints) in G is $\Theta(n)$, using methods from Chapter 8 of [2], it is not hard to verify that the degree of each vertex (variable) is, asymptotically, a Poisson distributed random variable with mean a constant, for both options for Step 1. This implies that for any constant d there almost surely exist $\Omega(n)$ vertices that have degree exactly 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. 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.

Finally, 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 probability of having at least D incompatible pairs and the same arguments, as above, can be applied.

3.1 A new model for random CSP

We would like to emphasize that "fixing" the old models by conditioning on each value having degree less than D in C^{Π}, will probably not lead to any interesting new models. This is because the old models generate many other *local* flaws similar to the one in the previous section. For example,we could have a pair of variables where neither is impossible to set by itself but where every setting of one makes it impossible to set the other (again for some local reasons as for the previous problem).

Regarding 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 and that in fact the models may remain problematic for $p_2 < 1/D$. What would be, perhaps, more important and certainly more practically relevant, is shifting from constraints that contain an entirely random subset of $p_2 D^2$ forbidden pairs to constraints were 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 new model that is simple to state and analyze, similarly to the currently used models, which does not suffer from trivial local inconsistencies. For this model 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) for a random CSP instance Π.

Definition 1 (Model E) *C^Π is as a random n-partite graph with D nodes in each part constructed by uniformly, independently and with repetitions selecting $m = p\binom{n}{2}D^2$ edges out of the $\binom{n}{2}D^2$ possible ones.*

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 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 1 *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 1 is well known for $D = 2$ (see [9], where a proof is given in terms of 2-SAT) and having $D > 2$ only makes things easier for that proof. Moreover,

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

Fact 2 is the analog of Theorem 5e in [11] when edges are selected with repetitions. By fact 2 and fact 1 we get

Theorem 1 *If for a random instance Π generated using Model E, C^Π has fewer than $\frac{n}{2}$ edges then Π almost surely has a solution.*

We feel that substantially stronger bounds can be derived by a more sophisticated analysis of this model and we leave this as future work.

4 Bounding the overconstrained region

4.1 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 decreases exponentially with n. Such a condition is readily provided by the *first moment* method (for excellent expositions see [2] and [24]). 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,

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

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.

4.2 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". The technique introduced in [19], when applied to CSP, 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 Pi, if we choose a variable at random and change its value, the probability we will end up with a non-solution is rather small. 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 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 [17]. Hence, the technique could potentially apply to other models for random CSP. This technique has already been applied to random SAT [19] and random graph coloring [1], in each case giving the best bounds currently known.

Definition 2 *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. *any 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 1 $\Pr[\Pi \text{ has a solution}] \leq E[|\mathcal{S}_n^\sharp|]$.

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 by conditioning on $A \in \mathcal{S}_n$:

Lemma 2 $E[|\mathcal{S}_n^\sharp|] = \Pr[A \in \mathcal{S}_n] \sum_{A \in \mathcal{A}_n} \Pr[A \in \mathcal{S}_n^\sharp \mid A \in \mathcal{S}_n]$.

5 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 2 *For a randomly generated binary CSP instance Π, if $(1 - \frac{1}{D^2})^r \frac{1-\zeta^D}{1-\zeta} < 1$, where $\zeta = 1 - e^{-\frac{2r}{D^2-1}}$, then $\Pr[\Pi$ has a solution$]$ tends to 0 as n tends to infinity.*

Proof As we argued earlier, $\Pr[A \in \mathcal{S}_n] = (1 - \frac{1}{D^2})^m$. We will compute an upper bound to $\Pr[A \in \mathcal{S}_n^\sharp \mid A \in \mathcal{S}_n]$. Fix a value assignment $A \in \mathcal{S}_n$. Since $A \in \mathcal{S}_n$, every pair of values of A is *not* adjacent (incompatible) in C. Consequently, conditioning on $A \in \mathcal{S}_n$ implies that the set of possible edges 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,

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

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 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 fact follows from the main Theorem in [21]. This implies,

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

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 1 and 2, we get:

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

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 2 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 3 *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.*

6 The method of local maxima for model B

We will now use the method of local maxima in order to derive better estimations for the probability of a random CSP instance having a solution for a popular model in Artificial Intelligence, called Model B in [22, 23]. As we showed in section 3, this model suffers from the shortcoming that when n tends to infinity the probability that a randomly generated instance has a solution tends to zero. However, when n is fixed and relatively small, it is unlikely that there will arise a variable with all the values in its domain flawed for many interesting values of the defining parameters (we know this from experiments!). We illustrate this with the following example:

Consider, for the sake of simplicity in the calculations of the example and only for this paragraph, the model where each constraint edge is selected independently with probability p_1 and for each selected constraint edge, each incompatibility edge is selected in the same way with probability p_2 (Model A, in [22, 23]) and suppose we have n_0 variables. Let v be a fixed value in the domain of a fixed variable X. Each constraint that bounds X makes v flawed with probability p_2^D. Hence, if X is bound by c constraints, the probability that v is flawed is $1-\left(1-p_2^D\right)^c$ and the probability that all the values in the domain of X are flawed is $\left(1-\left(1-p_2^D\right)^c\right)^D$. Even if we are very generous and take $c=n_0$ for $n_0=200, D=10, p_2=1/2$ the expected number of variables that have all the values in their domain flawed is less than 10^{-5} and, hence, so is the probability that such a vertex exists. So, the following estimate which improves over the first moment one, might be useful in suggesting fruitful parameter ranges for conducting experiments with Model B.

Recall that for an instance Π, G is generated by first selecting $m_1=p_1\binom{n}{2}=cn$ constraints and C is constructed by selecting $m_2=p_2D^2$ incompatible pairs (edges in C) for each constraint edge in G. Consider a value assignment A that is a solution of Π and a change of some variable X in A to a value v greater than X's value in A. We count the possible edge occurrences in C that would imply $A(X,v)\not\models\Pi$ and divide them by the total number of edges than can be present under the condition that $A\models\Pi$.

$A(X,v)\not\models\Pi$ iff at least one of the m selected constraints involves X and another variable, and contains an edge connecting v and the value assigned to the variable by A. The number of "bad" edges is $(n-1)\binom{D^2-1}{m_2-1}$, while conditioning on $A\models\Pi$ implies that the set of possible edges has cardinality $\binom{n}{2}\binom{D^2-1}{m_2}$. Therefore, for each of the m_1 selected constraint edges of Π, the probability that $A(X,v)$ is not a solution, given $A\models\Pi$, is:

$$\frac{(n-1)\binom{D^2-1}{m_2-1}}{\binom{n}{2}\binom{D^2-1}{m_2}}=\frac{2m_2}{n(D^2-m_2)}$$

implying

$$\Pr[A(X,v)\not\models\Pi\mid A\in\mathcal{S}_n]=1-\left(1-\frac{2p_2}{n(1-p_2)}\right)^{m_1}.$$

Arguing similarly to the proof of Theorem 2 we can show that the events $A(X_i,v_j)\not\models\Pi$ are negatively correlated for each X_i and v_j, getting

$$\Pr[A \in \mathcal{S}_n^\sharp \mid A \in \mathcal{S}_n] \leq \left(1 - \left(1 - \frac{2p_2}{n(1-p_2)}\right)^{m_1}\right)^{sf(A)}. \tag{5}$$

Again, 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 1 and 2 and setting $\zeta = 1 - \left(1 - \frac{2p_2}{n(1-p_2)}\right)^{m_1}$, we get:

$$\begin{aligned}
&\Pr[\Pi \text{ has a solution}] \\
&\leq (1-p_2)^{m_1} \sum_{A \in \mathcal{A}_n} \Pr[A \in \mathcal{S}_n^\sharp \mid A \in \mathcal{S}_n] \\
&\leq (1-p_2)^{m_1} \sum_{k_1, k_2, \ldots, k_D} \binom{n}{k_1, k_2, \ldots, k_D} \prod_{j=1}^{D} \left(1 - \left(1 - \frac{2p_2}{n(1-p_2)}\right)^{m_1}\right)^{(D-j)k_j} \\
&= (1-p_2)^{m_1} \left(\sum_{j=0}^{D-1} \left(1 - \left(1 - \frac{2p_2}{n(1-p_2)}\right)^{m_1}\right)^j\right)^n \\
&= (1-p_2)^{m_1} \left(\frac{1-\zeta^D}{1-\zeta}\right)^n .
\end{aligned}$$

Acknowledgements 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," *Manuscript,* University of Toronto.

[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] 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.

[12] A. El Maftouhi and W. Fernandez de la Vega, "On random 3-SAT", *Combinatorics, Probability and Computing,* Vol. 4 (1995) 189–195.

[13] E. Friedgut, "Necessary and sufficient conditions for sharp thresholds," *Manuscript in preparation.*

[14] 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.

[15] A. Goerdt, "A threshold for satisfiability," in I.M. Haven and V. Koubek (eds.) *Proceedings of the Symposium on the Mathematical Foundations of Computer Science* (1992) 264–274.

[16] T. Hogg, "Refining the phase transition in combinatorial search," in [3], 127–154.

[17] 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.

[18] 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.

[19] L. M. Kirousis, E. Kranakis, and D. Krizanc, "Approximating the unsatisfiability threshold of random formulas," *Proceedings of the Fourth Annual European Symposium on Algorithms, ESA '96,* (1992) 27–38.

[20] A.K. Mackworth, "Constraint satisfaction," in S. Shapiro (ed.) *Encyclopedia of Artificial Intelligence*, Wiley, New York, 2nd ed. (1992) 285–293.

[21] C. McDiarmid, "On a correlation inequality of Farr," *Combinatorics, Probability and Computing* Vol. 1 (1992) 157–160.

[22] P. Prosser, “An empirical study of phase transitions in binary constraint satisfaction problems,” in [3], 81–109.

[23] B.M. Smith and M.E. Dyer, “Locating the phase transition in binary constraint satisfaction problems,” in [3], 155-181.

[24] J. H. Spencer, *Ten Lectures on the Probabilistic Method*, 2nd edition, SIAM, Philadelphia (1994).

[25] D. Waltz, “Understanding line drawings of scenes with shadows,” *The Psychology of Computer Vision*, McGraw-Hill, New York (1975) 19–91.

[26] C. Williams and T. Hogg, “Using deep structure to locate hard problems,” *Proceedings of AAAI-92* (1992) 472–477.

[27] C. Williams and T. Hogg, “Extending deep structure,” *Proceedings of AAAI-93*, (1993) 152–158.

[28] C. Williams and T. Hogg, “Exploiting the deep structure of constraint problems,” *Artificial Intelligence*, Vol. 70 (1994) 73–117.