When does the giant component bring unsatisfiability?

Michael Molloy*
Department of Computer Science
University of Toronto
Toronto, Canada

March 22, 2005

Abstract

We study random constraint satisfaction problems using the wide class of models introduced by the author[36], which includes various forms of random SAT and other well-studied problems. We determine precisely which of these models remain almost surely satisfiable when the number of clauses is increased beyond the point at which a giant component appears in the underlying constraint hypergraph.

1 Introduction

A constraint satisfaction problem (CSP) is a generalization of a boolean formula in conjunctive normal form. Roughly speaking, a CSP generalizes SAT in the sense that variables can draw their values from a more general domain than simply $\{T, F\}$, and each clause (a.k.a. constraint) consists of a set of restrictions as to which values the variables in that clause may jointly take.

In [36], the author introduced a general class of models for random constraint satisfaction problems. It includes and generalizes many well-studied random problems, such as random k-SAT (see eg, [1, 2, 8, 10, 15, 23, 29, 31, 33]) random NAE-SAT[3] and the colourability of random graphs [4, 6, 7, 9, 18, 22, 32]. The same class, in a slightly less general form, was introduced independently by Creignou and Daudé[16]. See [36, 5, 24] for a further introduction to this class of models, including a description of how its study was necessitated by a very active, but misled, study of thresholds for random constraint satisfaction problems conducted by many researchers in Artificial Intelligence.

In this class of models, one always begins by randomly selecting k-tuples of variables on which to place a constraint. Then, for each k-tuple, one chooses a random constraint. The distribution from which this random constraint is chosen is what specifies the model. For one such distribution, the model is random k-SAT, for another it is d-colourability of random graphs, etc. We explain this more precisely in the next subsection.

*This work is supported by an NSERC Research Grant and a Sloan Research Fellowship. Much of this work was done while the author was a Visiting Researcher at Microsoft Research.

The most important situations arise when there are n variables and cn constraints, where c is some absolute constant. Generally, one wishes to determine for which values of c the random CSP is almost surely[1] (a.s.) satisfiable and for which it is a.s. unsatisfiable. Of course, this depends on which model from the class is being studied. For example, in the case of random 3-SAT, we know that the random problem is a.s. satisfiable for $c < 3.42$[31] and a.s. unsatisfiable for $c > 4.506$[19]. For random 3-colourability, the random problem is a.s. satisfiable, (i.e. the random graph is a.s. 3-colourable) for $c < 2.01$[9] and a.s. unsatisfiable for $c > 2.4682$[18]. (In both cases, it is conjectured that the upper and lower bounds can be improved to the point where they coincide.)

The main focus of [36] was to answer the following questions:

(Q1) Which models have the property that there exists $c_1 > 0$ such that the random problem is a.s. satisfiable for all $c < c_1$?

(Q2) Which models have the property that there exists $c_2 > 0$ such that the random problem is a.s. unsatisfiable for all $c > c_2$?

Answering Q2 turned out to be trivial. We also answered Q1, but the answer was somewhat unsatisfactory for the following reason. The underlying constraint hypergraph of the random CSP undergoes a remarkable change at $c = 1/k(k-1)$. For $c < 1/k(k-1)$, all of its components are very small, most are trees, and none have more than 1 cycle. For $c > 1/k(k-1)$, it has a giant component on $\Theta(n)$ vertices which has many cycles. So answering Q1 with $c_1 = 1/k(k-1)$ required simply determining which models are a.s. satisfiable when the random CSP is the union of several problems whose constraint hypergraphs are very small and trivial. This indicates that the following is a much better question:

(Q3) Which models have the property that there exist $c_1 > 1/k(k-1)$ such that the random problem is a.s. satisfiable for all $c < c_1$?

In other words, for which models is the random problem a.s. satisfiable for some values of c which are large enough that there is a.s. a giant component?

In this paper, we answer Q3. The author was pleasantly surprised to discover that, with only a particular contrived group of exceptions (specified in Section 3), every model satisfying the property demanded in Q1 also satisfies the stronger property demanded in Q3.

One reason for the importance of answering Q3 is that doing so determines those models whose threshold is non-trivial in the sense that the threshold occurs in (the interior of) a range of c where the problems are not trivially satisfiable. Another reason is that this contributes to our understanding of the kinds of impact that the first giant component to appear can have on the random structure. In particular, we see that with regards to properties that can be phrased in the form "is the random CSP satisfiable" (eg. d-colourability), the giant component has surprisingly little impact.

A formal statement of our main result will have to wait until after a lengthy set of definitions. For now, we describe it informally using Figure 1. In each of the plots in that figure, the x-axis is c, and the y-axis is the limit as n tends to infinity of the probability of satisfiability. There is a dashed horizontal line indicating $c = 1/k(k-1)$, the point where the giant component a.s. appears. So in every diagram, the random CSP remains a.s. satisfiable until the giant component threshold. In Plot A, it remains a.s. satisfiable for some values of c which exceed the giant component threshold, i.e. the answer to Q3 is YES. The other 3 plots illustrate what might happen when the answer is NO. In Plot D, there is a

[1]We say that a property P holds almost surely if the probability of P tends to 1 as n tends to infinity.

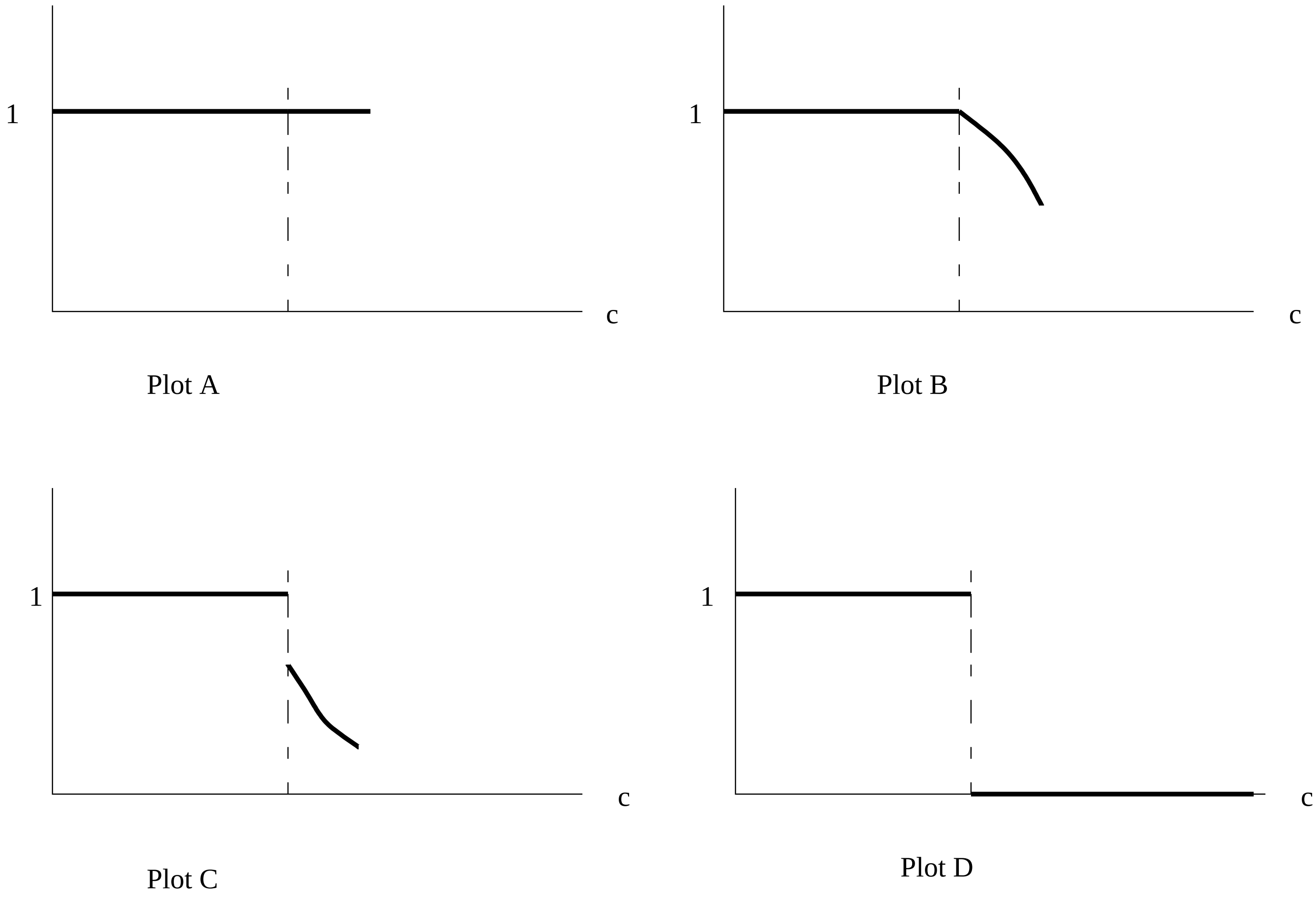

Figure 1: Four possible plots of the probability of satisfiability. The dashed line indicates the value of c at which the giant component a.s. appears.

sharp threshold of satisfiability at $c = 1/k(k-1)$; i.e. the CSP is a.s. unsatisfiable for every c above the giant component threshold. In the other two, it ceases to be a.s. satisfiable at the giant component threshold, but for some values of c greater than that threshold, it is not a.s. unsatisfiable either.

We prove that the situation represented by the Plot C never occurs. We completely characterize for which models each of the other 3 situations occurs. In doing so, we prove that if the set of permissable constraints is "connected", a natural property defined in Section 3, then only the situation represented by the first plot can occur; i.e. the answer to question Q3 is always YES.

As an example, it is well known that the satisfiability threshold for 2-SAT is $c = 1$ [14, 21, 25] (also see [13]). The underlying hypergraph for 2-SAT is the usual random graph $G_{n,M=cn}$ and so the threshold for it to have a giant component is $c = \frac{1}{2}$. Thus, these thresholds do not coincide. Interestingly, the satisfiability threshold for 2-SAT does coincide with the threshold for the underlying graph to have a giant *strongly connected* component, if it is viewed as a directed graph [30], and also with the threshold for a different underlying graph, one whose vertices are the literals rather than the variables, to have a giant component.

We will also discuss the possible situations for models where the random problem is not a.s. satisfiable for $c < 1/k(k-1)$.

2 The random model

Here, we recall from [36] the definition of the random model.

In our setting, the variables of our problem all have the same domain of permissable values, $\mathcal{D} = \{1, ..., d\}$, and all constraints will have size k, for some fixed integers d, k. Given a k-tuple of variables, $(x_1, ..., x_k)$, a *restriction* on $(x_1, ..., x_k)$ is a k-tuple of values $R = (\delta_1, ..., \delta_k)$ where each $\delta_i \in \mathcal{D}$. A set of restrictions on a k-tuple $(x_1, ..., x_k)$ is called a *constraint*. The *empty constraint* is the constraint which contains no restrictions. A *constraint satisfaction problem* (CSP) consists of a domain-size d, a constraint-size k, a collection of variables, and a set of constraints on k-tuples of those variables.[2] We say that an assignment of values to the variables of a constraint C *satisfies* C if that assignment is not one of the restrictions in C. An assignment of values to all variables in a CSP satisfies that CSP if every constraint is simultaneously satisfied. A CSP is *satisfiable* if there is at least one such satisfying assignment.

Given a CSP F, a *sub-CSP* of F is a CSP obtained by removing some of the variables and constraints from F, with the rule that if a variable is removed then all constraints involving that variable must also be removed. The sub-CSP *induced* by a subset V_1 of the variables of F is the CSP on V_1 containing every constraint of F that involves only variables of V_1.

The *constraint hypergraph* of a CSP is the k-uniform hypergraph whose vertices correspond to the variables, and whose hyperedges correspond to the k-tuples of variables which have *constraints*. Of course, when $k = 2$, the constraint hypergraph is simply a graph, and so we often call it the *constraint graph*.

It will be convenient to consider a set of canonical variables $X_1, ..., X_k$ which are used only to describe the "pattern" of a constraint. These canonical variables are not variables of the actual CSP. For any d, k there are d^k possible restrictions and 2^{d^k} possible constraints over the k canonical variables. We denote this set of constraints as $\mathcal{C}^{d,k}$. For our random model, one begins by specifying a particular probability distribution, $\mathcal{P}$ over $\mathcal{C}^{d,k}$. Different choices of $\mathcal{P}$ give rise to different instances of the model, with varying threshold behaviours.

The Random Model: Specify M, n and $\mathcal{P}$ (typically $M = cn$ for some constant c; note that $\mathcal{P}$ implicitly specifies d, k). First choose a random constraint hypergraph with M hyperedges, in the usual manner; i.e., where each k-uniform hypergraph with n vertices and M hyperedges is equally likely. Next, for each hyperedge e, we choose a constraint on the k variables of e as follows: we take a random permutation from the k variables onto $\{X_1, ..., X_k\}$ and then we select a random constraint according to $\mathcal{P}$, mapping it onto a constraint on our k variables in the obvious manner.

We use $CSP_{n,M}(\mathcal{P})$, to denote a random CSP drawn from this model with parameters $n, M, \mathcal{P}$. We occasionally omit the subscript n, M, depending on the context. We say that a property holds *almost surely* (a.s.) if the probability that it holds tends to 1 as n tends to infinity.

[2] Actually, CSP's are often defined to be much more general than this. For example, the domains might change from variable to variable and the sizes of the constraints may differ.

As described in [36], this generalizes many well studied models of random structures. For example, different choices of $\mathcal{P}$ make $CSP_{n,M}(\mathcal{P})$ to be equivalent to random k-SAT, random NAE-k-SAT, and colourability of random graphs/hypergraphs. Creignou and Daude [16] independently introduced and studied a model for generalized satisfiability problems which reduces to a special case of our model with the restriction that $\mathcal{P}$ is always the uniform distribution on a subset of $\mathcal{C}^{2,k}$.

As mentioned in [36], we could have chosen the constraint hypergraph by making an independent choice for each potential hyperedge, deciding to put it in the hypergraph with probability $p = \frac{c \times k!}{n^{k-1}}$. We use $CSP_{n,p}(\mathcal{P})$ to denote such a random CSP. $CSP_{n,p}(\mathcal{P})$ is, in many senses, equivalent to the model described above. In particular, it is easy to show that all the theorems in this paper translate to this alternate model. We will make use of this equivalence in the proofs of Lemmas 6 and 7.

3 Definitions and the main theorem for $k = 2$

The bulk of this paper will deal with the case $k = 2$. The case $k \geq 3$ will be deferred to the final section. For now, we remark that the case $k \geq 3$ essentially reduces to the case $k = 2$.

Most of these definitions are taken from [36], sometimes rephrased for the special case $k = 2$.

We will always use $\mathcal{C}$ to denote a set of constraints on the canonical variables X_1, X_2.

For any constraint C on X_1, X_2, the *reflection* of C, denoted C^{-1}, is the constraint whose restrictions are the set $\{(y, x) : (x, y) \text{ is a restriction of } C\}$. In other words, this can be thought of as the constraint obtained from C by exchanging X_1 and X_2. We say that a set $\mathcal{C}$ is *reflective* if for every constraint $C \in \mathcal{C}$, the reflection of C is also in $\mathcal{C}$. $\mathcal{P}$ is *reflective* if supp$(\mathcal{P})$ is reflective and if every constraint C has the same probability as its reflection.

Remark: Note that when studying our random model, we can always assume without loss of generality that $\mathcal{P}$ is reflective. This is because our model takes a random permutation of the actual variables of the CSP to X_1, X_2 each time it chooses a random constraint. So for any constraint $C \in \text{supp}(\mathcal{P})$, if we add its reflection to $\mathcal{P}$ and assign both it and C a probability of half the original probability of C, we do not affect the distribution of the random CSP's generated by the model.

Given a set $\mathcal{C}$ of constraints on X_1, X_2, we say that a constraint on any non-canonical variables y_1, y_2 *belongs* to $\mathcal{C}$ if upon replacing each y_i by X_i we obtain a constraint in $\mathcal{C}$. We say that a particular CSP f can be *formed* from $\mathcal{C}$ if all the constraints of f belong to $\mathcal{C}$.

A *tree CSP*, *cyclic CSP*, *unicyclic CSP* is one whose constraint graph is respectively a tree, a cycle, a connected graph with exactly one cycle.

A constraint C *forbids* $X_i = \delta$ if each of the d pairs (x_1, x_2) with $x_i = \delta$ is a restriction of C. Such a C is a *forbidding constraint.*

We define *bad* values of the domain recursively as follows: Given a set $\mathcal{C}$, we say that a value δ is *0-bad*, if there is some canonical variable X_i, and constraint $C \in \mathcal{C}$ such that C forbids $X_i = \delta$. We say that δ is *j-bad* if there is some canonical variable X_i, and constraint $C \in \mathcal{C}$ such that C implies that if $X_i = \delta$ then the other canonical variable must be assigned a j'-bad value for some $j' < j$. A value is *bad* if it is j-bad for some j. A value is *good* if it is not bad.

We consider the following properties that $\mathcal{C}$ might have:

Property 1: For every $\delta \in \mathcal{D}$, there is at least one $C \in \mathcal{C}$ such that $X_1 = \delta, X_2 = \delta$ does not satisfy C.

Property 2: There is at least one good value in $\mathcal{D}$.

Property 3: Every cyclic CSP formed from $\mathcal{C}$ has a satisfying assignment which uses only good values.

Note that $\mathcal{C}$ must have at least one of Properties 1 and 2, since if some δ violates the condition of Property 1, then δ is a good value. If $\mathcal{C}$ does not satisfy Property 1 for some δ, then clearly every CSP formed from $\mathcal{C}$ is trivially satisfied by setting every variable equal to δ. The following lemma motivates Properties 2 and 3.

Lemma 1 *(a) $\mathcal{C}$ can form an unsatisfiable tree CSP iff it does not satisfy Property 2.*

(b) If $\mathcal{C}$ satisfies Property 2, then it can form an unsatisfiable unicyclic CSP iff it does not satisfy Property 3.

Proof This was proved in [36]. Specifically, proving (a) took up most of the proof of Theorem 4 and proving (b) took up most of the proof of Theorem 5. □

$G_{n,M}$ is the random k-uniform hypergraph on n vertices formed by choosing M uniformly random edges independently without replacement. A fundamental result from random graph theory (see [20] or [28]) says that if $c > \frac{1}{2}$ then a.s. $G_{n,M=cn}$ has a giant component on $\Theta(n)$ vertices and if $c < \frac{1}{2}$ then a.s. all the components of $G_{n,M=cn}$ are of size $O(\log n)$ and have a fairly simple structure. Using this, and other basic properties of random graphs, the author [36] showed:

1. If supp($\mathcal{P}$) does not satisfy Property 1, then for every $c > 0$, $CSP_{n,M=cn}(\mathcal{P})$ is satisfiable.
2. If supp($\mathcal{P}$) does not satisfy Property 2, then for every $c > 0$, $CSP_{n,M=cn}(\mathcal{P})$ is a.s. unsatisfiable.
3. If supp($\mathcal{P}$) satisfies Properties 1, 2 but not 3 then for every $0 < c < \frac{1}{2}$, $CSP_{n,M=cn}(\mathcal{P})$ is neither a.s. satisfiable nor a.s. unsatisfiable and for some $c \geq \frac{1}{2}$, $CSP_{n,M=cn}(\mathcal{P})$ is a.s. unsatisfiable.
4. If supp($\mathcal{P}$) satisfies Properties 1, 2 and 3 then for every $0 < c < \frac{1}{2}$, $CSP_{n,M=cn}(\mathcal{P})$ is a.s. satisfiable and for some $c \geq \frac{1}{2}$, $CSP_{n,M=cn}(\mathcal{P})$ is a.s. unsatisfiable.

Since $\mathcal{C}$ must satisfy at least one of Properties 1 and 2, this list covers all possibilities for $\mathcal{C}$.

Items 3 and 4 of this characterization are somewhat unsatisfactory in that they merely say that for such $\mathcal{P}$, $CSP(\mathcal{P})$ is not a.s. unsatisfiable (resp. is a.s. satisfiable) so long as the constraint graph has a fairly trivial structure. The main theorems of this paper rectify that shortcoming. Theorem 2 determines precisely which distributions $\mathcal{P}$ are such that $CSP(\mathcal{P})$ remains a.s. satisfiable when c is increased past the point where the constraint graph a.s. has a giant component. Theorem 4 determines which $\mathcal{P}$ remain not a.s. unsatisfiable when c is increased past that point.

A set $\mathcal{C}$ is defined in [36] to be *well-behaved* if it satisfies Properties 1 and 2 and to be *very-well-behaved* if it satisfies Properties 1, 2 and 3. The primary goal of this paper is to better understand those distributions $\mathcal{P}$ which fall under item 4, and so we will mainly focus on $\mathcal{P}$ such that supp($\mathcal{P}$) is very-well-behaved.

A set of constraints $\mathcal{C}$ on X_1, X_2 yields a directed graph $G(\mathcal{C})$ with vertex set $\mathcal{D}$ and with the edge v_1, v_2 present iff there is some constraint $C \in \mathcal{C}$ which permits $X_1 = v_1, X_2 = v_2$. $\mathcal{C}$ is said to be *connected* if the undirected graph obtained by removing directions from the edges of $G(\mathcal{C})$ is connected. We say $\mathcal{C}$ is *strongly connected* iff $G(\mathcal{C})$ is strongly connected, i.e. for every $\delta_1, \delta_2 \in \mathcal{D}$ there is a directed path from δ_1 to δ_2. A component of that undirected graph is said to be a *component* of $\mathcal{C}$. A strongly connected component of $G(\mathcal{C})$ is said to be a *strongly connected component* of $\mathcal{C}$. The *value set* of a (strongly connected) component of $\mathcal{C}$ is the set of values which make up the vertex set of that (strongly connected) component in the corresponding graph. The *(strongly connected) components* of a constraint C, are the (strongly connected) components of $\mathcal{C} = \{C\}$. We sometimes say $G(C)$ instead of $G(\{C\})$. Note that if $\mathcal{C}$ is reflective then $\mathcal{C}$ is connected iff it is strongly connected.

A constraint C_2 *contains* a constraint C_1 (denoted $C_1 \subseteq C_2$) if C_2 permits all the pairs of values that C_1 does (and possibly some additional pairs). In other words, if $G(C_1)$ is a subgraph of $G(C_2)$. $\mathcal{C}_2$ *subsumes* $\mathcal{C}_1$ (denoted $\mathcal{C}_1 \trianglelefteq \mathcal{C}_2$) if for every constraint $C_1 \in \mathcal{C}_1$ there is a constraint $C_2 \in \mathcal{C}_2$ such that $C_1 \subseteq C_2$.

We say that a constraint C is an *expanded permutation* if it has the following structure: There is a partition $\mathcal{V}_C = V_1, ..., V_t$ of $\{1, ..., d\}$ along with a permutation $\phi = \phi_C$ of $\{1, ..., t\}$ such that (a, b) is permissible in C iff for some i, $a \in V_i$ and $b \in V_{\phi(i)}$. In other words, the edge set of $G(C)$ is simply the union over all i of all edges from V_i to $V_{\phi(i)}$. In this case, we say that $V_1, ..., V_t$ is the *underlying partition* of C. A set of constraints $\mathcal{C}$ is a *permutation set* if (i) every $C \in \mathcal{C}$ is an expanded permutation; (ii) these expanded permutations all have the same underlying partition, i.e. if for all $C, C' \in \mathcal{C}$, we have $\mathcal{V}_C = \mathcal{V}_{C'}$; and (iii) that underlying partition has at least 2 parts, i.e. $t \geq 2$. In this case, we can say that this partition is also the *underlying partition* for $\mathcal{C}$. We say that $\mathcal{C}$ is a *subpermutation set* if it is subsumed by a permutation set $\mathcal{C}^*$. The *underlying partition* for $\mathcal{C}^*$ is also said to be an *underlying partition* for $\mathcal{C}$. (Note that an underlying partition of a subpermutation set is not necessarily unique.)

Given a constraint C on the canonical variables X_1, X_2 and the value set $\{1, ..., d\}$, and given any subset $V \subseteq \{1, ..., d\}$, the *projection* C_V of C to V is the constraint on X_1, X_2 with domain V consisting of all the restrictions $(\delta_1, \delta_2) \in C$ for which δ_1, δ_2 are both in V. In other words, $G(C_V)$ is the subgraph of $G(C)$ induced by V. Thus, if one were to reduce the domain of a CSP to V in the obvious manner, each constraint would be replaced by its projection. The *projection* of $\mathcal{C}$ to V is the set of projections to V of the elements of $\mathcal{C}$. Thus, the projection $\mathcal{C}_V$ of $\mathcal{C}$ to V is a set of constraints on X_1, X_2 with domain V. Observe that $G(\mathcal{C}_V)$ is a subgraph of $G(\mathcal{C})$, but is not necessarily the subgraph induced by V.

Suppose that the value-sets of the components of $\mathcal{C}$ are $V_1, V_2, ..., V_t$ and let $\mathcal{C}_i$ be the projection of $\mathcal{C}$ to V_i. We say that $\mathcal{C}_i$ is a *component* of $\mathcal{C}$. (In this case, $G(\mathcal{C}_i)$ is equal to the component of $G(C)$ on V_i.) Thus $\mathcal{C}_i$ is a set of constraints with domain set V_i, and so definitions such as well-behaved apply to the components of $\mathcal{C}$.

Before stating our main theorem formally, we will give an informal description. Recall that we are focussing our attention on $\mathcal{P}$ such that supp($\mathcal{P}$) is very-well-behaved, i.e. satisfies Properties 1,2,3. For such $\mathcal{P}$, a.s. we can satisfy the constraints of all non-giant components of $CSP_{n,M=cn}(\mathcal{P})$, and so the main question is to determine whether we can satisfy the constraints of the giant component. Note that all variables in the giant component must receive values from the value set of a particular component of $\mathcal{C} = \text{supp}(\mathcal{P})$.

If $\text{supp}(\mathcal{P})$ is connected, then we will prove that we get Plot A of Figure 1. (Proving that fact is the most substantial part of this paper.) If at least one component $\mathcal{C}_i$ is very-well-behaved, then by only considering values from V_i, it is as if we were in the case where $\text{supp}(\mathcal{P})$ is connected and very-well-behaved, and so we also get Plot A of Figure 1.

In the case that no component of $\text{supp}(\mathcal{P})$ is very-well-behaved, then for each component $\mathcal{C}_i$, we can use its constraints to form an unsatisfiable CSP which is either a tree CSP or a unicyclic CSP (by Lemma 1). If that CSP occurs as a sub-CSP of the giant component, then of course, the giant component cannot be satisfied using the values from $\mathcal{C}_i$. It is not hard to show that, with probability bounded away from 0, the giant component will have one such sub-CSP for each of the components, and thus it will be unsatisfiable. So we do not get Plot A of Figure 1.

If no component of $\text{supp}(\mathcal{P})$ is very-well-behaved then we look for the next best thing, which turns out to be a component $\mathcal{C}_i$ which is not a subpermutation set and which has at least one good value. The constraints of such a component cannot form an unsatisfiable tree CSP, and the unsatisfiable unicyclic CSP's that it can form are few enough that, with probability bounded away from 0, the giant component won't have any as sub-CSP's. If it does not have any such sub-CSP's then (as we will prove) we can satisfy the giant component using the values from $\mathcal{C}_i$. Thus, we get Plot B of Figure 1.

If no component of $\text{supp}(\mathcal{P})$ is very-well-behaved, and if every component $\mathcal{C}_i$ with at least one good value is a subpermutation set, then every $\mathcal{C}_i$ can form either an unsatisfiable tree CSP, or an abundance of unsatisfiable unicyclic CSP's. In each case, the giant component will a.s. have one of these unsatisfiable sub-CSP's. Thus a.s. the giant component can't be satisfied using values from $\mathcal{C}_i$. Since this is true for each component of $\text{supp}(\mathcal{C})$, a.s. the giant component is unsatisfiable. Thus, we get Plot D of Figure 1.

Plot C of Figure 1 never occurs.

The formal statement of our main theorem is as follows. Case (a) is represented by Plot A in Figure 1, case (b) is Plot D and case (c) is Plot B.

Theorem 2 *Consider any $d \geq 2$ and any distribution $\mathcal{P}$ on $\mathcal{C}^{d,2}$. Suppose that $\text{supp}(\mathcal{P})$ is very-well-behaved.*

(a) If $\text{supp}(\mathcal{P})$ has a very-well-behaved component then there exists $\epsilon > 0$ such that for every $c \leq \frac{1}{2}+\epsilon$, $CSP_{n,M=cn}(\mathcal{P})$ is a.s. satisfiable.

(b) Else if every component of $\text{supp}(\mathcal{P})$ with at least one good value is a subpermutation set, then for every $c > \frac{1}{2}$, $CSP_{n,M=cn}(\mathcal{P})$ is a.s. unsatisfiable.

(c) Else then there exists $\epsilon > 0$ such that for every $\frac{1}{2} < c \leq \frac{1}{2} + \epsilon$ there exists $\gamma_1 = \gamma_1(\mathcal{P}, c) < 1$ and $\gamma_2 = \gamma_2(\mathcal{P}, c) > 0$ such that

$$\gamma_1 + o(1) \geq \Pr(CSP_{n,M=cn}(\mathcal{P})\text{is satisfiable}) \geq \gamma_2 + o(1).$$

Furthermore, $\gamma_2(\mathcal{P}, c)$ approaches 1 as c approaches $\frac{1}{2}$ from above.

Thus, for each $\frac{1}{2} < c \leq \frac{1}{2} + \epsilon$, $CSP_{n,M=cn}(\mathcal{P})$ is neither a.s. satisfiable nor a.s. unsatisfiable.

Note that if $\text{supp}(\mathcal{P})$ is connected and very-well-behaved, then it falls into hypothesis (a) of Theorem 2. Thus our theorem implies, perhaps surprisingly, that for connected $\text{supp}(\mathcal{P})$, the appearance of the

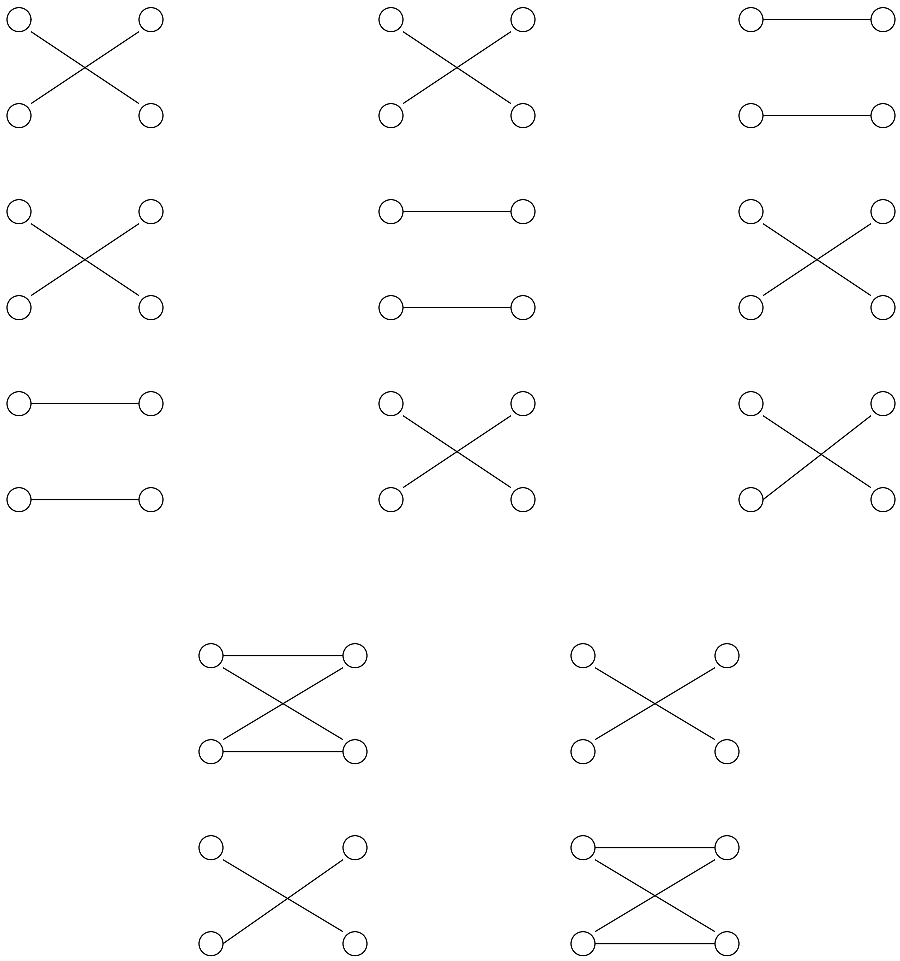

Figure 2: Here, all pairs of *permitted* values are joined by lines. The first set of three constraints on domain $\{1,...,6\}$ falls under case (b) of Theorem 2. The second set, on domain $\{1,...,4\}$ falls under case (c).

giant component never coincides with a jump from $o(1)$ to $\Theta(1)$ in the probability that $CSP_{n,M=cn}(\mathcal{P})$ is unsatisfiable:

Corollary 3 *There is no $\mathcal{P}$ with* $\mathrm{supp}(\mathcal{P})$ *connected, such that* $\Pr(CSP_{n,M=cn}(\mathcal{P})$ *is unsatisfiable*$) = o(1)$ *for all $c < \frac{1}{2}$ and* $\Pr(CSP_{n,M=cn}(\mathcal{P})$ *is unsatisfiable*$) = \Theta(1)$ *for all $c > \frac{1}{2}$.*

It is not immediately obvious that there is any choice of $\mathrm{supp}(\mathcal{P})$ which falls into case (b) of Theorem 2. Henry Cohn provided me with such an example, and it is shown in Figure 2. It is easier to come up with examples which fall into case (c); one such example is given in the same figure.

3.1 A proof outline

The bulk of the proof is for the case that $\mathcal{C}$ is connected. For this case, we introduce a structure called a "semi-null constraint-path" and prove that if $\mathcal{C}$ is very-well-behaved and can form a semi-null constraint-path then there exists $\epsilon > 0$ such that for $c \leq \frac{1}{2}+\epsilon$, $CSP_{n,M=cn}(\mathcal{P})$ is a.s. satisfiable (Lemma 6). Next we prove that every connected $\mathcal{C}$ can form a semi-null constraint-path. To do so, we show that if a connected $\mathcal{C}$ cannot form a semi-null constraint-path, then $\mathcal{C}$ must be a subpermutation set (Lemma 17). Finally, we show that every permutation set can form an unsatisfiable cyclic CSP (Lemma 19) and note that this implies that so can every subpermutation set. Therefore, $\mathcal{C}$ must be able to form a semi-null constraint-path or else we contradict the hypothesis that $\mathcal{C}$ is very-well-behaved.

For the disconnected case, we note that in any satisfying assignment, variables in the same component of the underlying graph must have values in the same component of $\mathcal{C}$. Then we argue about the probability that the variables in the giant component of the underlying hypergraph can be assigned values from V_i without violating any of the constraints in $\mathcal{C}_i$ for at least one component $\mathcal{C}_i$ of $\mathcal{C}$. Since each $\mathcal{C}_i$ is connected, this relies heavily on the machinery built up to handle the aforementioned case where $\mathcal{C}$ is connected.

In the next section, we introduce null and semi-null constraint-paths, and prove various related lemmas. We also introduce a few other important concepts and tools. In the following section, we develop several lemmas concerning collections of expanded permutations. Those two sections contain most of the labour. In Section 6, we use these results to prove Theorem 2 and Theorem 4 (which is stated in the next subsection). Finally, in Section 7 we extend Theorems 2 and 4 to general $k \geq 3$.

3.2 Not very-well-behaved supp($\mathcal{P}$).

We close this section by discussing the case where $\mathcal{C} = \text{supp}(\mathcal{P})$ is not very-well-behaved. If it is not even well-behaved then, as proved in [36], $CSP_{n,M=cn}(\mathcal{P})$ is a.s. unsatisfiable for every $c > 0$ or it is a.s. satisfiable for every $c > 0$, and so there is not much to say. So we focus on the case where $\mathcal{C}$ is well-behaved but not very-well-behaved, i.e. where it satisfies Properties 1 and 2 but not 3. Thus, for $0 < c < \frac{1}{2}$, $CSP_{n,M=cn}(\mathcal{P})$ is neither a.s. satisfiable nor a.s. unsatisfiable.

The analogue of Figure 1 is Figure 3. We will prove that Plot B of Figure 3 can never occur and if $\mathcal{C}$ is connected then only Plots A and D can occur.

Theorem 4 *Consider any $d \geq 2$ and any distribution $\mathcal{P}$ on $\mathcal{C}^{d,2}$. Suppose that* supp($\mathcal{P}$) *is well-behaved but not very-well-behaved. Then there is a continuous, monotonically decreasing function $\alpha : [0, \frac{1}{2}] \rightarrow [0,1]$ with $\alpha(0) = 1$ such that for $0 \leq c < \frac{1}{2}$:*

$$\mathbf{Pr}(CSP_{n,M=cn}(\mathcal{P})\text{is satisfiable }) = \alpha(c) + o(1).$$

Furthermore:

- *(a) If each component of* supp($\mathcal{P}$) *with at least one good value is a subpermutation set then for every $c \geq \frac{1}{2}$, $CSP_{n,M=cn}(\mathcal{P})$ is a.s. unsatisfiable. If, in addition,* supp($\mathcal{P}$) *is connected then $\alpha(\frac{1}{2}) = 0$.*
- *(b) Else:*

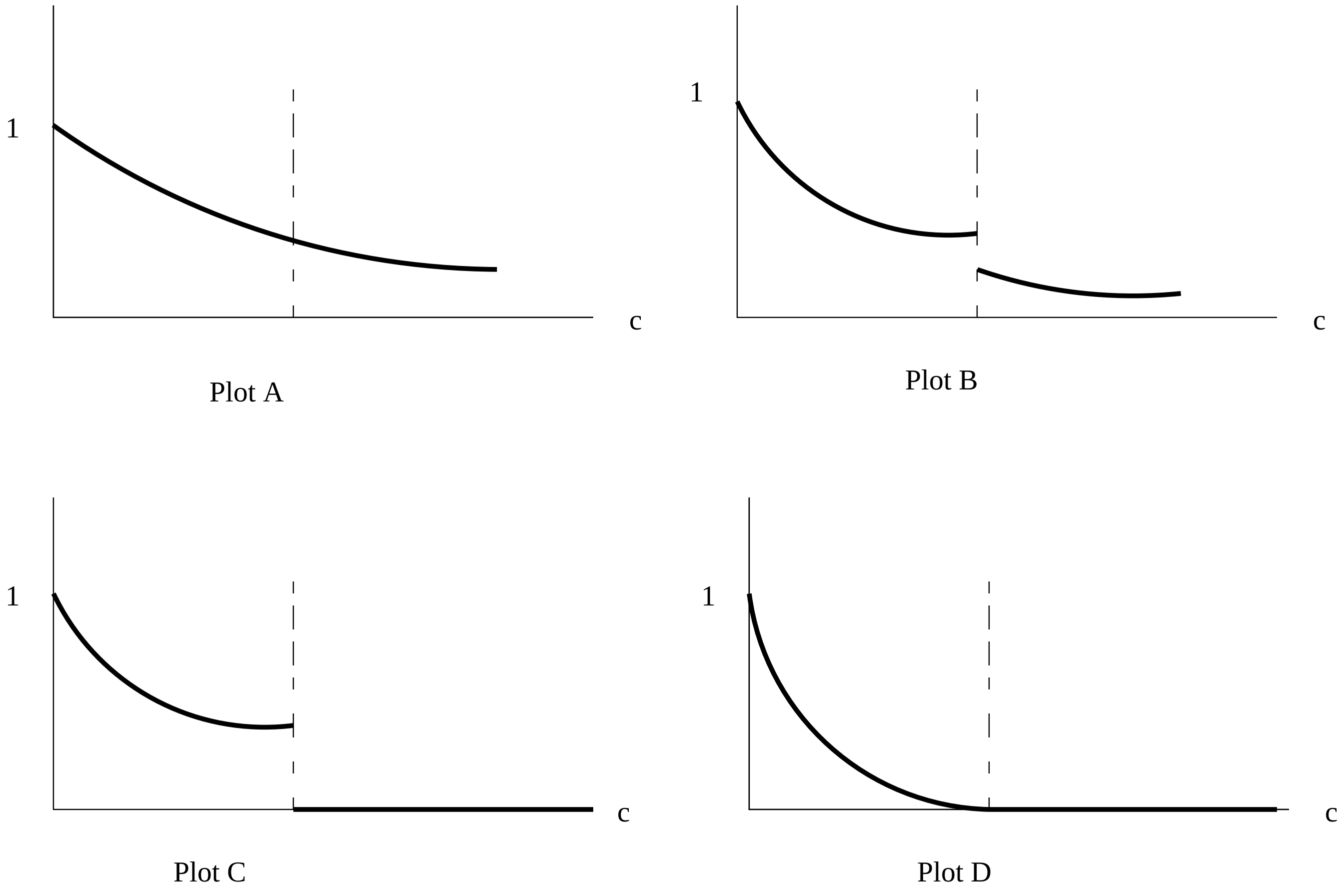

Figure 3: Four possible plots of the probability of satisfiability. The dashed line indicates the value of c at which the giant component a.s. appears.

(i) $\alpha(\frac{1}{2}) > 0$*; and*

(ii) There exists $\epsilon > 0$ *such that for every* $\frac{1}{2} < c \leq \frac{1}{2} + \epsilon$ *there exists* $\gamma_2 = \gamma_2(c) > 0$ *such that*

$$\Pr(CSP_{n,M=cn}(\mathcal{P}) \text{ is satisfiable}) \geq \gamma_2 + o(1).$$

Furthermore, $\gamma_2(c)$ *approaches* $\alpha(\frac{1}{2})$ *as* c *approaches* $\frac{1}{2}$ *from above.*

Thus, for each $0 < c \leq \frac{1}{2} + \epsilon$, $CSP_{n,M=cn}(\mathcal{P})$ *is neither a.s. satisfiable nor a.s. unsatisfiable.*

Less formally, in situation (b) we have Plot A of Figure 3. In situation (a), if $\text{supp}(\mathcal{P})$ is connected, then we must have Plot D; if $\text{supp}(\mathcal{P})$ is disconnected then we have either Plot C or D.

Theorem 4, along with Corollory 3, yields:

Corollary 5 *If* $\text{supp}(\mathcal{P})$ *is connected and well-behaved, then we do not have* $\lim_{n\to\infty} \Pr(CSP_{n,M=cn}(\mathcal{P})$ *is unsatisfiable) dropping suddenly from being at least some constant* $\alpha > 0$ *for every* $c < \frac{1}{2}$ *to* $o(1)$ *for every* $c > \frac{1}{2}$.

Examples where we obtain Plots A and D are easy to produce. It's less obvious how to produce an example for Plot C; one is provided in Figure 4. It is a modification of the first example from Figure 2, where domain values 5,6 from that figure are each replaced by a set of 4 values, and another constraint is added to make the collection reflective. For $c > \frac{1}{2}$ a.s. $CSP_{n,M=cn}(\mathcal{P})$ is unsatisfiable, for essentially the same reason as for the first example from Figure 2. The only way to form an unsatisfiable cycle with these constraints is to take 5 copies of the first constraint, or symmetrically, 5 copies of the second constraint. So for $c < \frac{1}{2}$, $\alpha(c)$ is the limit of the probability that $CSP_{n,M=cn}(\mathcal{P})$ does not have one of these two cycles, which is at least the probability that $G_{n,M=cn}$ has no cycles - a probability that is well known to be at least some positive constant (see eg. [12]).

4 Preliminaries

In this section, we present the key definitions and lemmas which are required to prove our main theorem. Throughout this section, we will assume that $k = 2$, i.e. that every constraint is on only two variables.

4.1 Null and semi-null constraint-paths

We begin by defining the null constraint-path, which is one of the key concepts of this paper. A substantial part of the work in this paper will be to characterize which sets $\mathcal{C}$ can form such paths.

A *constraint-path* of $\mathcal{C}$ with length ℓ is a sequence of variables $v_0, v_1, ..., v_\ell$ with a constraint from $\mathcal{C}$ between each pair v_{i-1}, v_i (where v_{i-1} corresponds to X_1 and v_i corresponds to X_2). We call v_0, v_ℓ the endpoints of the constraint-path and $v_1, ..., v_{\ell-1}$ the internal variables. (These variables must be distinct; in particular, we do not allow $v_0 = v_\ell$.) We say that such a constraint-path is *formed* by $\mathcal{C}$. A constraint-path of length ℓ *permits* (x, y) if there is some set of assignments to $v_1, ..., v_{\ell-1}$ which, along with setting $v_0 = x, v_\ell = y$, does not violate any of the constraints in the path. A constraint-path of length ℓ is *null* if it permits (x, y) for every pair x, y in the domain.

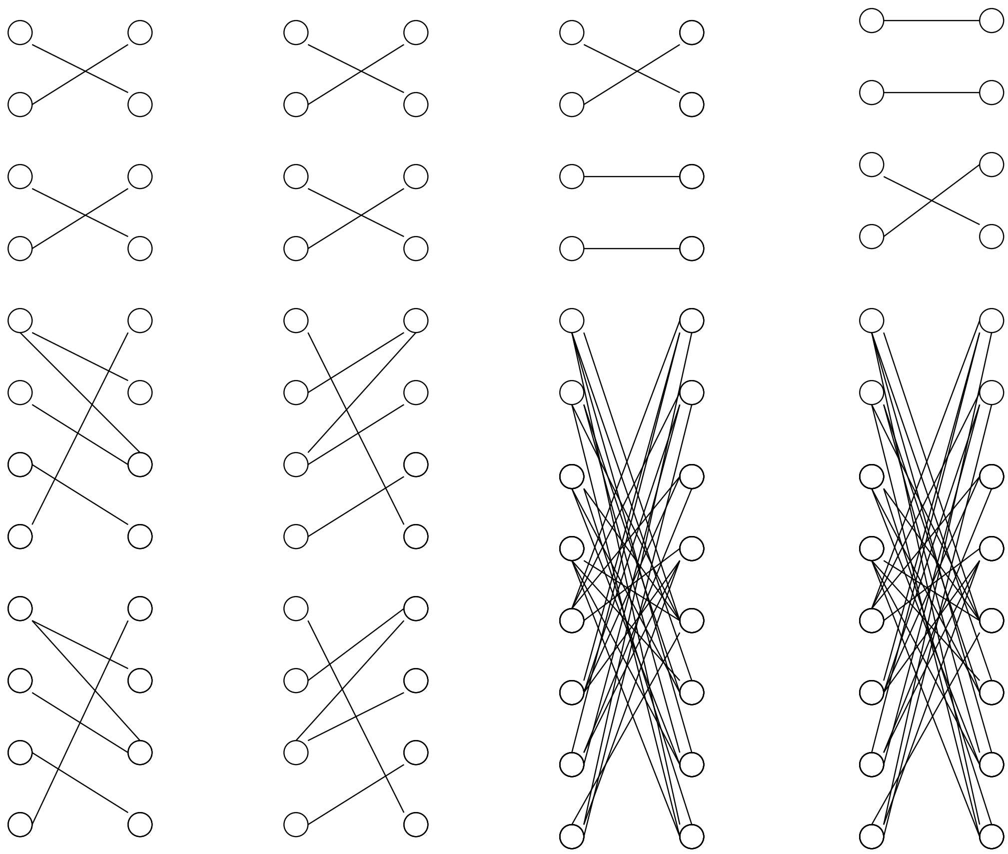

Figure 4: All pairs of *permitted* values are joined by lines. The domain is $\{1, ..., 12\}$. In the last two constraints, all pairs in $\{5, 6, 7, 8\} \times \{9, 10, 11, 12\}$ and $\{9, 10, 11, 12\} \times \{5, 6, 7, 8\}$ are permitted.

Bad values can prevent $\mathcal{C}$ from being able to form a null constraint-path. For example, suppose that for every constraint $C \in \mathcal{C}$, there is some value δ such that C forbids $X_1 = \delta$. Then it is not possible for C to form a null constraint-path, as for every path P, there is some δ such that the first constraint of P forbids the first variable from taking δ, and so P does not permit (δ, y) for any $y \in \mathcal{D}$. This behooves us to relax the notion of null constraint-path as follows:

We define a *semi-null* constraint-path to be a constraint-path P such that P permits (x, y) for every pair of good values x, y.

The following lemma shows the importance of such constraint-paths.

Lemma 6 *If* $\text{supp}(\mathcal{P})$ *can form a semi-null constraint-path then there exists* $\epsilon > 0$ *such that for* $c \leq \frac{1}{2} + \epsilon$*:*

(a) if $\text{supp}(\mathcal{P})$ *is very-well-behaved then* $CSP_{n,M=cn}(\mathcal{P})$ *is a.s. satisfiable;*

(b) if $\text{supp}(\mathcal{P})$ *has at least one good value then there exists* $\gamma > 0$ *such that* $\mathbf{Pr}(CSP_{n,M=cn}(\mathcal{P})\text{is satisfiable}) \geq \gamma + o(1)$.

Proof summary: Given a random formula f, we consider a subformula $\Psi(f)$ defined as follows: for every semi-null path P in f such that the internal variables do not lie in any clauses outside of P, we remove P, i.e. we remove its clauses and its internal variables. We show that a.s. $\Psi(f)$ can be satisfied using only good values. Any such assignment can be extended to a satisfying assignment for f.

We expose $\Psi(f)$ using a variation on a standard branching process. If we were exposing f, then the branching process would have positive drift, but the fact that some paths are deleted is enough to give the branching process on $\Psi(f)$ a negative drift. It follows that a.s. all of its components are unicyclic and if $\text{supp}(\mathcal{P})$ is very-well-behaved, then we can argue that all such components can be satisfied using only good values. If $\text{supp}(\mathcal{P})$ is not very-well-behaved, but has at least one good value, then all tree components can be satisfied using only good values. With probability $\gamma > 1$ all components of $\Psi(f)$ will be trees. $\square$

Recall that for $c > \frac{1}{2}$, the underlying constraint graph of $CSP_{n,M=cn}(\mathcal{P})$ will a.s. have a giant component. The next lemma bounds the probability of unsatisfiability for the sub-CSP induced by that giant component; i.e. the sub-CSP consisting of all the variables of that component and all the constraints from $CSP_{n,M=cn}(\mathcal{P})$ which use those variables.

Lemma 7 *If* $\text{supp}(\mathcal{P})$ *has at least one good variable and can form a semi-null constraint-path then for every* $\xi > 0$ *there exists* $\epsilon > 0$ *such that for* $c = \frac{1}{2} + \epsilon$*, the probability that the sub-CSP induced by the largest component of* $CSP_{n,M=cn}(\mathcal{P})$ *is unsatisfiable is less than* ξ*.*

Proof summary: We choose $f, \Psi(f)$ as in the proof of Lemma 6. The size of the giant component of f tends to 0 with ϵ. A.s. $\Psi(f)$ has very few cyclic components, and so for sufficiently small ϵ, the probability that no cyclic component of $\Psi(f)$ lies in the giant component of f is at least $1 - \xi$.

Lemma 8 *If* $\text{supp}(\mathcal{P})$ *is a connected reflective permutation set then for every* $\epsilon > 0$ *there exists* $c < \frac{1}{2}$ *such that* $\mathbf{Pr}(CSP_{n,M=cn}(\mathcal{P})$ *is satisfiable*$) < \epsilon$*.*

We postpone the proof of Lemma 8 until the end of Section 5. For now, we note a simple extension:

Corollary 9 *If* supp$(\mathcal{P})$ *is a connected reflective subpermutation set then for every* $\epsilon > 0$ *there exists* $c < \frac{1}{2}$ *such that* $\mathbf{Pr}(CSP_{n,M=cn}(\mathcal{P})$ *is satisfiable*$) < \epsilon$.

Proof We consider a distribution $\mathcal{P}^*$ defined as follows. We start with $\mathcal{P}^*(C) = 0$ for all C and then for each $C \in \text{supp}(\mathcal{C})$ we choose some $C^* \in \mathcal{C}^*$ such that $C \subseteq C^*$ and add $\mathcal{P}(C)$ to $\mathcal{P}^*(C^*)$ and $\mathcal{P}(C^{-1})$ to $\mathcal{P}^*((C^*)^{-1})$ (recall that C^{-1} is the reflection of C). Note that supp$(\mathcal{P}^*)$ is a connected reflective permutation set. Clearly, $\mathbf{Pr}(CSP_{n,M=cn}(\mathcal{P})$ is satisfiable$) < \mathbf{Pr}(CSP_{n,M=c^*n}(\mathcal{P}^*)$ is satisfiable$)$ since we can select a CSP from the distribution $CSP_{n,M=c^*n}(\mathcal{P}^*)$ by first taking one from $CSP_{n,M=cn}(\mathcal{P})$ and replacing each constraint by one that contains it; this replacement cannot introduce new satisfying assignments. So the corollary follows from Lemma 8. □

4.2 The proofs

Here we give the full proofs for Lemmas 6 and 7. The proof of Lemma 8 appears at the end of Section 5.

Proof of Lemma 6 Our proof will use a slight twist on standard branching process arguments.

Let P be some semi-null constraint-path formed by supp$(\mathcal{P})$ and let ℓ be its length. Suppose that we were to build a constraint-path of length ℓ by taking any sequence of $\ell + 1$ variables and then choosing the ℓ constraints independently at random from the distribution $\mathcal{P}$. Let $q > 0$ be the probability that this constraint-path would be P.

We will work in the $CSP_{n,p}(\mathcal{P})$ model with $p = 2c/n$; it is straightforward to show that the result then translates to $CSP_{n,M=cn}(\mathcal{P})$. So choose some f from $CSP_{n,p}(\mathcal{P})$. We will expose f, by starting at a particular variable v and doing something resembling a breadth-first search of the constraint graph rooted at v.

We say that an *ear* is a path $v_0, ..., v_t$ or cycle $v_0, v_1, ..., v_t = v_0$ in the constraint graph such that $\deg(v_1) = \deg(v_2) = ... = \deg(v_{t-1}) = 2$ and $\deg(v_0), \deg(v_t) \neq 2$. v_0, v_t are the *endpoints* of the ear; $v_1, ..., v_{t-1}$ are the *internal variables* (so if an ear is a cycle, v_0 counts as both endpoints). Note that an edge is an ear, with $t = 1$, if neither endpoint has degree 2. Note also that every vertex of degree 2 lies either in a (unique) ear or an isolated cycle, i.e. a cycle where every vertex has degree exactly 2 and thus which forms a component of the graph.

We will consider the sub-CSP $\Psi(f)$ obtained from f by removing every ear which forms the constraint-path P, i.e. removing its constraints and internal variables from f (this rule implies that no ears which are cycles are removed). We will show that, with sufficiently high probability, $\Psi(f)$ has a satisfying assignment which uses only good values. This will suffice, since to extend that assignment to f, one needs only to assign values to the internal vertices of the ears of $f - \Psi(f)$ so as not to violate any of the constraints in those ears. Since these ears are all semi-null constraint-paths, we can do this regardless of which good values are assigned to their endpoints.

For any particular vertex v, we will expose the component of $\Psi(f)$ containing v using the following procedure, which is a slight variation on standard branching process procedures commonly used on random graphs.

1. Initially, $U = \{v\}$, $W = \emptyset$; U, W are the sets of *unexplored* and *explored* variables in the component of $\Psi(f)$ containing v.
2. Expose the degree of v; if it is 2, then v is an internal variable of an ear or isolated cycle and so we must perform a special preprocessing step:
   (a) Expose the ear or isolated cycle that v is an internal variable of.
   (b) If it is an isolated cycle, then HALT. That cycle is the component of $\Psi(f)$ that contains v.
   (c) If it is an ear that forms P, then HALT. In this case $v \notin \Psi(f)$.
   (d) Else, place the endpoints of the ear into U, remove v from U, and place v and all the other internal variables of the ear into W.
3. While $U \neq \emptyset$, choose an arbitrary $u \in U$ and:
   (a) Expose all the ears for which u is an endpoint.
   (b) For each such ear that does not form P: place all the internal variables of the ear into W; if the other endpoint of the ear is not in $U \cup W$, then place it into U.
   (c) Remove u from U, and place it into W.

Note that we never add an internal variable of an ear to U. When we stop, W contains all the variables in the component of $\Psi(f)$ that contains v.

Remark: A typical branching process argument exposes only the edges out of u. Our twist is to expose the ears.

We let U_i be the size of U after i iterations of the main loop. The expected number of ears out of u is at most p times the number of variables which have not yet been exposed in our process, and that product is at most $2c + o(1)$. The probability that an ear is removed is at least $q' + o(1)$ where $q' > 0$ is q times the probability that the ear has length ℓ. Therefore, at any step after the preprocessing step, the expected number of new variables added to U is at most $2c(1 - q') + o(1)$ which is less than 1 for $c < \frac{1}{2} + \epsilon$ with $\epsilon = \frac{1}{2(1-q')} - \frac{1}{2}$. Since U loses a variable at each step, U_i is a random walk with negative drift.

Standard arguments show that the expected number of variables that we explore before reaching $U_i = 0$, is $O(1)$. Furthermore, with probability greater than $1 - 1/n^2$, the graph induced by W, i.e. the component of $\Psi(f)$ containing v, will not contain two cycles. Finally, there is a $z > 0$, such that the probability that this component has a cycle is less than z/n. We omit the details which are straightforward for anyone who is familiar with branching process arguments in random graph theory (see, eg Section 5.2 of [28]).

After reaching $U_i = 0$, we continue to expose the rest of the components of $\Psi(f)$ by removing the variables in W, and all constraints containing them, from the formula. We also remove any ears that we exposed which formed P. In other words, we delete the component of $\Psi(f)$ that contains v and any ears forming P joining that component to the rest of $\Psi(f)$ (in the case that v was not in $\Psi(f)$ then we simply delete the ear forming P that contains v.) Then we start the process again at $U = \{v'\}$ with another arbitrary variable v'. Again, the probability that the component of $\Psi(f)$ containing v' has two cycles is less than $1/n^2$ and the probability that it contains one cycle is less than z/n.

After repeating this process at most n times, we will have exposed every component of $\Psi(f)$. With probability at least $(1-1/n^2)^n = 1-o(1)$, each of these components has at most one cycle, and with probability at least $(1-z/n)^n = \mathrm{e}^{-z}+o(1)$, they are all trees.

We now use a stronger version of Lemma 1; it's proof is implicit in the proofs of Theorems 4 and 5 of [36]:

Claim: If supp($\mathcal{P}$) is very-well-behaved, then every tree CSP and unicyclic CSP formed from supp($\mathcal{P}$) can be satisfied using only good values. If, supp($\mathcal{P}$) has at least one good value, then every tree CSP formed from supp($\mathcal{P}$) can satisfied using only good values.

Therefore, if supp($\mathcal{P}$) is very-well-behaved, then a.s. $\Psi(f)$ can be satisfied using only good values, and as described earlier, such an assignment can be extended to a satisfying assignment of f. This proves part (a). If supp($\mathcal{P}$) has at least one good value then with probability at least $(1-z/n)^n = \mathrm{e}^{-z}+o(1)$, $\Psi(f)$ can be satisfied using only good values. This proves part (b) with $\rho = \mathrm{e}^{-z}$. □

Proof of Lemma 7 Suppose that f is a random CSP drawn from $CSP_{n,M=cn}(\mathcal{P})$. Again, we consider the branching process used in the proof of Lemma 6 to expose the components of $\Psi(f)$, the sub-CSP remaining after deleting every ear that forms a semi-null path. We say that a deleted such ear *touches* a component of $\Psi(f)$ if one of its endpoints lies in that component.

Claim: There exists $K > 0$ such that with probability at least $1-\xi/2$: every component of $\Psi(f)$ that contains a cycle has size at most K and is touched by fewer than K deleted ears.

The proof of this claim is straightforward for anyone experienced with branching process proofs for random graphs, and so we omit it.

Next, we prove that with probability at least $1-\xi/2$, no component of $\Psi(f)$ that has a cycle (i) is touched by at most K deleted ears, (ii) has size at most K, and (iii) lies in the giant component of f.

Choose an arbitrary vertex v and expose the component of $\Psi(f)$ containing v using the process described in the proof of Lemma 6. If that component (i) has a cycle, (ii) has size at most K, and (iii) is touched by at most K deleted ears, then expose the component of f containing v by setting U to be the other endpoints of those at most K deleted ears, (i.e. the endpoints that do not lie in the component of $\Psi(f)$ containing v), and then conducting K breadth-first searches from those endpoints.

As shown in the proof of Lemma 6, there is a constant x so that the probability that the component of $\Psi(f)$ which contains v has a cycle is at most x/n.

Those K breadth-first searches can be analyzed by straightforward branching processes. This time, since $c = \frac{1}{2}+\epsilon$, the branching processes have positive drift, i.e. the number of unexplored vertices tends to grow. Nevertheless, it is straightforward to show that, for ϵ sufficiently small in terms of ξ, there exists K_2 such that the probability that those K processes reach more than K_2 new vertices is at most $\xi/(2x)$. (Again, we omit the details.) Therefore, the probability that v lies in a cyclic component of $\Psi(f)$ and lies in a component of size greater than $K(\ell+1)+K_2$ of f is at most $\xi/(2n)$. (Recall that every deleted ear has exactly $\ell+1$ vertices, one of which is in the component of $\Psi(f)$ containing v.) Thus with probability at least $1-\xi/2$, there is no such vertex and hence no component of $\Psi(f)$ that contains a cycle (i) is touched by at most K deleted ears, (ii) has size at most K, and (iii) lies in the giant component of f.

Therefore, with probability at least $1-\xi$, every component of $\Psi(f)$ lying in the giant component of f is a tree and so the sub-CSP induced by the giant component of f is satisfiable. □

4.3 Generated constraints and strong connectivity

We say that a collection of constraints *generates* a constraint C if there is a constraint-path P whose constraints are all drawn from that collection such that for each pair $x, y \in D$, P permits (x, y) iff C does. We also say that C is *generated* by that path P. For example, $\mathcal{C}$ can form a null constraint-path iff $\mathcal{C}$ generates the empty constraint.

Recall that for any constraint C, C^{-1} is the reflection of C, i.e. for each $x, y \in \mathcal{D}$, C^{-1} permits (x, y) iff C permits (y, x). Recall that if $\mathcal{C}$ is *reflective* then for all $C \in \mathcal{C}$ we also have $C^{-1} \in \mathcal{C}$.

Given 2 constraints C_1, C_2, recall that C_1 *contains* C_2, denoted $C_2 \subseteq C_1$, if C_1 permits every pair (x, y) that C_2 permits. We use C_1C_2 to denote the constraint generated by the constraint-path of length 2 whose first constraint is C_1 and whose second constraint is C_2. We use C_1^ℓ to denote the constraint generated by the constraint-path of length ℓ whose constraints are all equal to C_1.

Proposition 10 *For any pair of constraints C_1, C_2 where C_1 does not forbid any values, we have $C_2 \subseteq C_1C_1^{-1}C_2$.*

Proof For every $x \in D$, $C_1C_1^{-1}$ permits (x, x) since x is not forbidden, and so if C_2 permits (x, y) then so does $C_1C_1^{-1}C_2$. □

Remark: Note that we don't necessarily have $C_2 = C_1C_1^{-1}C_2$ since $C_1C_1^{-1}$ can possibly permit some pairs (x, y) with $x \neq y$.

Recall that a set $\mathcal{C}$ yields a graph $G(\mathcal{C})$ with vertex set D and with the edge uv present iff there is some $C \in \mathcal{C}$ which permits (u, v), and that we say $\mathcal{C}$ is strongly connected if $G(\mathcal{C})$ is strongly connected, i.e. for every $\delta_1, \delta_2 \in D$ there is a directed path from δ_1 to δ_2. In other words, $\mathcal{C}$ is *strongly connected* if for every δ_1, δ_2 in the domain, there is some constraint path of $\mathcal{C}$ which permits (δ_1, δ_2). Note that different pairs δ_1, δ_2 may yield different paths of differing lengths, and so none of these paths is necessarily null.

5 Expanded permutations

The main contribution of this section is to show that if a reflective collection of constraints cannot form a semi-null constraint-path then it must be a subpermutation set. In doing so, we obtain a complete characterization of those reflective $\mathcal{C}$ that can form semi-null constraint-paths (Corollary 18) and of those that can form null constraint-paths for the case where $\mathcal{C}$ has no bad values (Corollary 12).

If $\mathcal{C}$ is a subpermutation set then it is easy to see that $\mathcal{C}$ cannot form a null constraint-path, since every constraint generated by $\mathcal{C}$ is an expanded permutation on the underlying partition.

Remark: If $\mathcal{C}$ is disconnected, then trivially it cannot form a null constraint-path as no constraint-path formed by $\mathcal{C}$ can permit (x, y) where x and y belong to different components of $\mathcal{C}$. Note that in this case, $\mathcal{C}$ is a subpermutation set - $\mathcal{C}^*$ has only one constraint, its underlying partition is the value sets of the components of $\mathcal{C}$, and ϕ is the identity permutation. So in a sense, subpermutation sets are a generalization of this most trivial class of structures that cannot form null constraint-paths.

The following lemma says that if $\mathcal{C}$ is reflective and has no bad values then being a subpermutation set is the only property that can prevent $\mathcal{C}$ from being able to form null-constraint paths.

Lemma 11 *Suppose that $\mathcal{C}$ is a collection of constraints, which is reflective and has no bad values. If $\mathcal{C}$ cannot form a null constraint-path then there exists a permutation set $\mathcal{C}^*$ such that:*

(i) every constraint in $\mathcal{C}^*$ can be generated from $\mathcal{C}$;

(ii) for every $C \in \mathcal{C}$ there exists $C^* \in \mathcal{C}^*$ such that $C \subseteq C^*$ (i.e. $\mathcal{C}$ is subsumed by $\mathcal{C}^*$); and

(iii) for every $C^* \in \mathcal{C}^*$ there exists $C \in \mathcal{C}$ such that $C \subseteq C^*$.

Corollary 12 *Suppose that $\mathcal{C}$ is a collection of constraints, which is reflective and has no bad values. Then $\mathcal{C}$ cannot form a null constraint-path iff $\mathcal{C}$ is a subpermutation set.*

Proof One direction follows immediately from Lemma 11. The other direction follows the same reasoning as that in the discussion preceding the lemma. To wit, if $\mathcal{C}$ is subsumed by such a collection, then for any constraint-path P induced by $\mathcal{C}$, and any domain-value i, P cannot permit both (i, j) and (i, k) if j, k lie in different parts of the underlying partition. $\square$

To prove Lemma 11, we require a series of lemmas.

Lemma 13 *For any constraint C with no forbidden values, we can use C, C^{-1} to generate a $C^{'}$ which contains C, such that every connected component of $C^{'}$ is strongly connected.*

Proof We say that a strongly connected component is *trivial* if it has no edges (i.e. if it has only one vertex and no loop). A strongly connected component is a *sink component* if there are no edges coming out of it, and a *source component* if there are no edges coming into it. Since C has no forbidden values, all sink components and source components are non-trivial.

Suppose a connected component B of C has more than one strongly connected component. Consider any source component A_1 of B. There is some path leading out of A_1; follow it to a sink component A_2. Suppose that it has k edges and goes from $a_1 \in A_1$ to $a_2 \in A_2$. Then we can show that $C_1 = C^k (C^{-1})^k C$ permits (a_2', a_1') for some $a_1' \in A_1, a_2' \in A_2$. To see this, note that since A_1, A_2 are non-trivial and strongly connected, there is some $a_1' \in A_1$ and $a_2' \in A_2$ such that C^k permits (a_2', a_2), $(C^{-1})^k$ permits (a_2, a_1) and C permits (a_1, a_1'). (Possibly $a_1' = a_1$ and/or $a_2' = a_2$.) Furthermore, since C has no forbidden values, neither does C^k and so by Proposition 10, $C \subseteq C_1$. Therefore, all values of A_1, A_2 lie in the same strongly connected component of C_1 and so C_1 has fewer strongly connected components than C. Repeating this operation a finite number of times will yield $C^{'}$, as desired. $\square$

Next, we show how to generate an expanded permutation that contains a particular constraint C.

Lemma 14 *For any constraint C with no forbidden values, we can use C, C^{-1} to generate an expanded permutation C^+ which contains C.*

Proof By Lemma 13, we can assume that every connected component of C is strongly connected.

We denote by P_k the constraint-path of length k in which all constraints are equal to C. For each pair of values x, y in the domain, $R(x, y)$ is the set of positive integers k such that P_k permits (x, y). Clearly, $R(x, x)$ is closed under addition. It follows that:

Claim 1: For each x there are positive integers $c(x), t(x)$ such that (i) $c(x)$ divides every member of $R(x,x)$ and (ii) $R(x,x)$ includes all multiples of $c(x)$ which are at least $t(x)$.

Proof: Note first that $|R(x)| = \infty$ since $R(x)$ is closed under addition, and since $R(x) \neq \emptyset$ as every connected component of C is strongly connected. Let c be the GCD of $R(x)$. Let $R' = \{i/c : i \in R(x)\}$. Since the GCD of R' is 1 and it is closed under addition, the Chinese Remainder Theorem implies that there is some t for which R' includes every integer greater than t. This proves the claim with $c(x) = c, t(x) = tc$.

Claim 1 quickly yields:

Claim 2: For each x, y in the same connected component of C, we have (i) $c(x) = c(y)$; (ii) all values in $R(x,y)$ are congruent mod $c(x)$; and (iii) there is some $t(x,y)$ such that $R(x,y)$ includes all members of that congruency class which are at least $t(x,y)$.

Proof: Since x, y are in the same connected component, and since that component is strongly connected, $R(x,y) \neq \emptyset$ and $R(y,x) \neq \emptyset$. If $i, j \in R(x,y)$ and $r \in R(y,x)$ then $i+r, j+r$ are both in $R(x,x)$, and so $i \equiv j (\text{mod } c(x))$. Similarly, if $i \in R(x,y), j \in R(y,x)$, and $r \in R(y,y)$ then $i+j$ and $i+j+r$ are both in $R(x,x)$, and so $r \equiv 0 (\text{mod } c(x))$. This implies that $c(x)$ divides $c(y)$ and, by the same argument, $c(y)$ divides $c(x)$. Thus $c(x) = c(y)$. Finally, if $i \in R(x,y)$ then for any $r \in R(x,x)$, we have $i + r \in R(x,y)$. Thus (iii) holds with $t(x,y) = i + t(x)$.

Taking $t = \max_{x,y}\{t(x,y), t(x)\} \times LCM(\{c(x) : x \in D\})$ we have:

Claim 3: (i) C^{t+1} contains C and (ii) every component of C^{t+1} is an expanded cyclic permutation.

Proof: (i) follows from the fact that C^t permits (x,x) for every x, and so C^tC permits (x,y) for every (x,y) permitted by C. To prove (ii), consider any x and for each $0 \leq i < c(x)$, let V_i be the set of values y such that $R(x,y) \neq \emptyset, R(y,x) \neq \emptyset$ and every element of $R(x,y)$ is congruent to i mod $c(x)$. For any $y \in V_0$, since $t+1 > t(x,y)$, we have $t+1 \in R(x,y)$ and so C^{t+1} permits (x,y). Similarly, C^{t+1} permits (a,b) for any $a \in V_i, b \in V_{i+1}$. Since these are the only possible edges between these sets, this proves (ii).

Setting $C^+ = C^{t+1}$ proves the lemma. □

Thus, we have a collection $\mathcal{C}^+$ of expanded permutations, each of which can be generated from the original members of $\mathcal{C}$, and such that every member of $\mathcal{C}$ is contained in a member of $\mathcal{C}^+$. All that remains to be shown is that we can obtain such a $\mathcal{C}^+$ in which all the members have the same underlying partition.

Lemma 15 *Consider any two expanded permutations C_1, C_2 on the same domain set with $\mathcal{V}_{C_1} = V_1, ..., V_t$ and $\mathcal{V}_{C_2} = U_1, ..., U_{t'}$, such that there exists $a, b \in V_1$ with $a \in U_1$ and $b \in U_2$. Then there is an expanded permutation C_3 such that:*

(i) C_3 can be generated by $\{C_1, C_2, C_1^{-1}, C_2^{-1}\}$;

(ii) $C_2 \subseteq C_3$;

(iii) $\mathcal{V}_{C_3}$ has fewer parts than $\mathcal{V}_{C_2}$.

Proof Consider $C' = C_1C_1^{-1}C_2$. By Lemma 14, we can use $C', (C')^{-1}$ to generate an expanded permutation C_3 which contains $C^{'}$, and hence (by Proposition 10) contains C_2. We just need to show that $\mathcal{V}_{C_3}$ has fewer parts than $\mathcal{V}_{C_2}$.

Case 1: U_1, U_2 lie in different cycles of C_2.

Since a, b are in the same part of $\mathcal{V}_{C_1}$, $C_1C_1^{-1}$ permits (a,a) and (a,b). Therefore, a and b are in the same connected component of $C^{'}$, and thus a and b are in the same cycle of the underlying permutation of C_3. Since C_3 contains C_2, it follows that the underlying permutation of C_3 has fewer cycles than that of C_2. Since no x lies in a cycle in the underlying permutation of C_3 which has longer length than the cycle in which x lies in the underlying permutation of C_2, it follows that $\mathcal{V}_{C_3}$ has fewer parts than $\mathcal{V}_{C_2}$.

Case 2: U_1, U_2 lie in the same cycle of C_2.

Since $C_1C_1^{-1}$ permits (a,b), $C^{'}$ permits (a,c) for any c lying in U_i, the part following U_2 in its cycle in the underlying permutation of C_2. Thus C^{-1}, and hence C_3, permits both (a,c) and (b,c) and so a, b are in the same part of C_3. Since no x, y in the same part of C_2 lie in different parts of C_3, this implies that $\mathcal{V}_{C_3}$ has fewer parts than $\mathcal{V}_{C_2}$. □

Corollary 16 *For every reflective collection of constraints, $\mathcal{C}$, with no bad values, there exists a collection of expanded permutations, $\mathcal{C}^*$ such that (i) every constraint in $\mathcal{C}^*$ can be generated from $\mathcal{C}$; (ii) for every $C \in \mathcal{C}$ there exists $C^* \in \mathcal{C}^*$ with $C \subseteq C^*$; and (iii) the members of $\mathcal{C}^*$ all have the same underlying partition.*

Remark: $\mathcal{C}^*$ is not necessarily a permutation set, because possibly the underlying partition has only one part.

Proof We use Lemma 14 to build a collection, $\mathcal{C}^+$, of expanded permutations which are generated from $\mathcal{C}$ such that for every $C \in \mathcal{C}$ there exists $C^+ \in \mathcal{C}^+$ with $C \subseteq C^+$. Suppose that $\mathcal{C}^+$ has two expanded permutations, C_1, C_2 which do not have the same underlying partition. Without loss of generality, we can assume that they are as in Lemma 15. Consider the collection obtained from C^+ by replacing C_2, C_2^{-1} with C_3, C_3^{-1}, where C_3 is the expanded permutation guaranteed by Lemma 15.

Clearly, the resulting collection is still reflective, and since C_3 can be generated from $\{C_1, C_1^{-1}, C_2, C_2^{-1}\}$, it can be generated from $\mathcal{C}$. Furthermore, since $C_2 \subseteq C_3$, we still have the property that every constraint of $\mathcal{C}$ is contained by a member of the resulting collection.

We repeat this process until the members of the collection all have the same underlying partition. At each step, we replace a constraint with one that has a smaller underlying partition, so we can only repeat a finite number of times. Thus, eventually we will terminate and obtain our collection $\mathcal{C}^*$. □

And now, Lemma 11 is a simple corollary:

Proof of Lemma 11: Corollary 16 implies that there is a $\mathcal{C}^*$ satisfying (i) and (ii). Condition (iii) is easily satisfied by removing from $\mathcal{C}^*$ any constraint that does not contain a member of $\mathcal{C}$; clearly this will not violate conditions (i) and (ii). If the underlying partition of $\mathcal{C}^*$ has only one part, then $\mathcal{C}^*$ has only one constraint: the null constraint. Since that constraint is generated from $\mathcal{C}$, by condition (i), then $\mathcal{C}$ can form a null constraint-path. □

Next, we extend Lemma 11 to the setting of semi-null constraint-paths. The only difference is that we do not obtain the analogue to condition (i) from Lemma 11.

Lemma 17 *Suppose that $\mathcal{C}$ is a reflective collection of constraints which cannot form a semi-null constraint-path. Then there exists a reflective permutation set $\mathcal{C}^*$ such that:*

(i) for every $C \in \mathcal{C}$ there exists $C^* \in \mathcal{C}^*$ such that $C \subseteq C^*$ (i.e. $\mathcal{C}$ is subsumed by $\mathcal{C}^*$); and

(ii) for every $C^* \in \mathcal{C}^*$ there exists $C \in \mathcal{C}$ such that $C \subseteq C^*$.

Proof The case where $\mathcal{C}$ is disconnected is easy: just let $\mathcal{C}^*$ consist of the single expanded permutation C which permits (x, y) iff x, y lie in the same component of $\mathcal{C}$. So we can assume that $\mathcal{C}$ is connected. Since it is reflective, it is strongly connected.

Let A be the set of good values in the domain. We can assume that $A \neq \emptyset$ as otherwise every constraint-path is semi-null and so the hypothesis is contradicted. Let $\mathcal{C}_A$ denote the projection of $\mathcal{C}$ onto A. We let $\mathcal{F}$ denote the set of constraints that can be generated from $\mathcal{C}$, and $\mathcal{F}_A$ denote the projection of $\mathcal{F}$ to A.

Since $\mathcal{C}$ is strongly connected, for every pair of values x, y there is some constraint in $\mathcal{F}$ which permits $X_1 = x, X_2 = y$. We will denote this as "permits (x, y)".

Claim: All members of A are good in $\mathcal{F}$.

Proof: Consider any constraint $C \in \mathcal{F}$. We have $C = C_1 C_2 ... C_r$ where each $C_i \in \mathcal{C}$. By definition, for every $y \in A$, and for every constraint C_i, there is some $z \in A$ such that C_i permits $X_1 = y, X_2 = z$. This implies that there is some $w \in A$ such that C permits $X_1 = x, X_r = w$. This in turn implies the claim.

Furthermore, the argument of the previous paragraph implies that all members of A are good in $\mathcal{F}_A$, and so $\mathcal{F}_A$ has no bad values, is connected, and is clearly reflective. Thus, by Lemma 11, either $\mathcal{F}_A$ can form a null constraint-path or there is a permutation set $\mathcal{F}_A^*$ such that (i) each constraint in $\mathcal{F}_A^*$ can be generated from $\mathcal{F}_A$, and (ii) for every $C \in \mathcal{F}_A$ there is a $C^* \in \mathcal{F}_A^*$ with $C \subseteq C^*$.

Claim: If $\mathcal{F}_A$ can form a null constraint-path then $\mathcal{C}$ can form a semi-null constraint-path.

Proof: Suppose P is a null constraint-path formed by $\mathcal{F}_A$, and that the constraints of P are $C_1, ..., C_r$ (in that order). Each C_i is the projection to A of some constraint $C_i' \in \mathcal{F}$. Consider the constraint-path P', formed by $\mathcal{F}$, whose constraints are $C_1', ..., C_r'$ (in that order). P' permits every pair that P does, and so P' is semi-null. Finally, every constraint of P' can be generated by $\mathcal{C}$ (by the definition of $\mathcal{F}$) and so P' can be formed from $\mathcal{C}$.

We now complete the proof of our lemma by showing that the existence of $\mathcal{F}_A^*$ implies the existence of the required $\mathcal{C}^*$.

Suppose that $\mathcal{F}_A^*$ has $t > 1$ parts: $\mathcal{V}_1, ..., \mathcal{V}_t$. We will create a partition $\mathcal{V}_1', ..., \mathcal{V}_t'$ of the entire domain, by adding each bad value to some $\mathcal{V}_i$. So, initially, $\mathcal{V}_i' = \mathcal{V}_i$ for each i.

Without loss of generality, suppose that $1 \in A$, and $1 \in \mathcal{V}_1$. Consider any value $x \notin A$. Since $\mathcal{C}$ is strongly connected, there is some constraint $C \in \mathcal{F}$ such that C permits $(x, 1)$. Because 1 is a good value of $\mathcal{C}$, our first Claim implies that it is also a good value of $\mathcal{F}$, and so C permits $(z, 1)$ for some good value z. Suppose $z \in \mathcal{V}_j$. Add x to $\mathcal{V}_j'$. We will show that this choice of j is well-defined for x.

Note that the existence of $\mathcal{F}_A^*$ ensures that C does not permit both $(z, 1)$ and $(z', 1)$ for any two good values $z \in \mathcal{V}_j, z' \in \mathcal{V}_{j'}$ with $j \neq j'$. Therefore, $\mathcal{V}_j$ is determined uniquely by our choice of C. So suppose that there are C_1, C_2 which give rise to different choices of j. That is, suppose that C_1 permits

$(x,1),(z_1,1)$ and C_2 permits $(x,1),(z_2,1)$ for two good values $z_1 \in \mathcal{V}_j$, $z_2 \in \mathcal{V}_{j'}$ with $j \neq j'$. Because z_1 is good, C_2 permits (z_1,w) for some good value w. Since z_1, z_2 are in different parts of the underlying partition of $\mathcal{F}_A^*$, and since C_2 permits (z_1,w) and $(z_2,1)$, we have that $w, 1$ must be in different parts. But, $C_1^{-1}C_2$ permits $(1,1)$ and $(1,w)$. This is a contradiction. Therefore, the choice of the part $\mathcal{V}_j'$ is uniquely determined for x. So our partition $\mathcal{V}_1', ..., \mathcal{V}_t'$ is well-defined, and furthermore:

Fact: Consider any $C \in \mathcal{F}$, and any bad value x such that C permits $(x,1)$. If z is a good value such that C permits $(z,1)$ then x, z are in the same part.

We next show that every constraint in $\mathcal{C}$ is contained by some expanded permutation on this partition. So consider any particular $C_1 \in \mathcal{C}$ and any part V_j'. Choose a particular good value $z \in V_j'$. Since j is good, C_1 permits (z,z') for some good value z'; suppose that $z' \in V_\ell'$. It will suffice to prove that for all $x \in V_j'$ if C_1 permits (x,x') then $x' \in V_\ell'$.

Suppose, to the contrary, that C_1 permits (x,x') for some $x \in V_j'$, $x' \in V_t'$ where $t \neq \ell$. The existence of $\mathcal{F}_A^*$ implies that at least one of x, x' is bad.

Case 1: x, x' are both bad. Since $\mathcal{C}$ is connected, there is some $C_2 \in \mathcal{F}$ that permits $(x',1)$. Since 1 is good, C_2 also permits $(y',1)$ for some good y'; our Fact implies that $y' \in V_t'$. Since y' is good, C_1 permits (y,y') for some good y. The existence of $\mathcal{F}_A^*$ implies that $y \notin V_j'$ since C_1 also permits (z,z') and z', y' are not in the same part. Therefore, C_1C_2 permits $(x,1)$ and $(y,1)$ where x is bad, y is good and x, y are in different parts. This contradicts our Fact.

Case 2: x is bad, x' is good. Since x' is good, C_1 permits (y,x') where y is good. Since x', z' are not in the same part, the existence of $\mathcal{F}_A^*$ implies that y, z are not in the same part; i.e. $y \notin V_j'$. Again, we consider some $C_2 \in \mathcal{F}$ that permits $(x',1)$. Thus, C_1C_2 permits $(x,1)$ and $(y,1)$. But x, y are not in the same part. This contradicts our Fact.

Case 3: x is good, x' is bad. Again, there is some $C_2 \in \mathcal{F}$ which permits $(x',1)$, and it follows as in Case 1 that C_1C_2 permits $(y,1)$ for some good $y \notin V_j'$. But C_1, C_2 also permits $(x,1)$. Since $x, y, 1$ are good and x, y are not in the same part, this contradicts the existence of $\mathcal{F}_A^*$.

This establishes the lemma by setting $\mathcal{C}^*$ to be the set of all expanded permutations on $\mathcal{V}_1', ..., \mathcal{V}_t'$ which contain at least one member of $\mathcal{C}$, thus ensuring that $\mathcal{C}^*$ satisfies condition (ii). Note that, since $\mathcal{C}$ is reflective, $\mathcal{C}^*$ is also reflective. □

Lemma 17 yields a complete characterization of those reflective collections $\mathcal{C}$ which cannot form a semi-null constraint-path:

Corollary 18 *Suppose that $\mathcal{C}$ is a reflective collection of constraints with at least one good value. Then $\mathcal{C}$ cannot form a semi-null constraint-path iff $\mathcal{C}$ is a subpermutation set.*

Remark: If $\mathcal{C}$ has no good values then every constraint-path that it forms is trivially semi-null.

Proof One direction follows immediately from Lemma 17. For the other direction, suppose $\mathcal{C}$ is a subpermutation set. Then for any constraint-path P induced by $\mathcal{C}$, and any domain-value i, P cannot permit both (i,j) and (i,k) if j, k lie in different parts of the underlying partition. Therefore, if P is a semi-null constraint path, it must be the case that all the good values lie in the same part of the underlying partition; call that part θ_1 and let θ_2 be any other part. By connectivity of $\mathcal{C}$, there is a constraint $C \in \mathcal{C}$ such that for any $i \in \theta_1$, C does not permit (i,j) for any $j \notin \theta_2$. Since every $j \in \theta_2$ is bad, this implies that every $i \in \theta_1$ is bad which contradicts the hypothesis that $\mathcal{C}$ has at least one good value. $\square$

Remark: If $\mathcal{C}$ has no good values then any path is trivially a semi-null constraint path. If $\mathcal{C}$ is disconnected and the good values are distributed over more than one component then $\mathcal{C}$ clearly cannot form a semi-null constraint-path. If $\mathcal{C}$ is disconnected and all good values lie in one component, then $\mathcal{C}$ can form a semi-null constraint-path iff the restriction of $\mathcal{C}$ to that component can. Thus, Corollary 18 provides a characterization for *all* reflective collections $\mathcal{C}$.

$\square$

Next we prove that if the collection $\mathcal{C}^*$ guaranteed by Lemmas 11 and 17 is connected, then it can form an unsatisfiable cyclic CSP.

Lemma 19 *If $\mathcal{C}^*$ is a connected reflective permutation set then $\mathcal{C}$ can form an unsatisfiable cyclic CSP.*

Proof Suppose that the underlying partition of $\mathcal{C}^*$ is $V_1, ..., V_t$ where $t \geq 2$.

We let G be the set of permutations of $[t]$ which can be obtained by taking the product of some of the permutations from the set $\phi_C : C \in \mathcal{C}^*$. Since $\mathcal{C}^*$ is reflective, for any $\phi \in G$, we also have $\phi^{-1} \in G$ and so G is a subgroup of the symmetric group S_t.

Since $\mathcal{C}^*$ is connected, G is transitive, i.e. for any $1 \leq i \leq j \leq t$, there is some permutation $\phi \in G$ mapping i to j.

For each $1 \leq i \leq t$, we let $G_i \subset G$ be the set of permutations in G for which i is a fixed point. Proposition 3.1 of [39] says that, since G is transitive, $\sum_{i=1}^{t} |G_i| = |G|$. (For completeness, we include the short proof of this proposition below.) Since the identity permutation lies in G_i for all i, and since $t > 1$, this implies that there is at least one permutation $\phi \in G$ which does not have any fixed points. Let k be the product of the lengths of the cycles of ϕ and let C be the expanded permutation produced by ϕ. It is easy to see that the cycle of $k+1$ copies of C is unsatisfiable.

We complete this proof with the proof of the key proposition.

It is easily seen that for each $1 \leq i \leq t$, G_i is a subgroup of G. Consider any $i > 1$ and some permutation $h \in G$ mapping 1 to i. For any $h' \in G$ mapping 1 to i, we have $h' = h'h^{-1}h$ and $h'h^{-1} \in G_1$. Therefore, the right coset $G_1 h$ of G_1 is equal to the set of permutations in G mapping 1 onto i and so there is a one-to-one correspondence between the right cosets of G_1 and the integers $1, ..., t$. Thus, G_1 has exactly t right cosets and so by Lagrange's Theorem, each has size exactly $|G|/t$, and in particular, $|G_1| = |G|/t$. The same argument yields $|G_i| = |G|/t$ for every $i \in [t]$, thus completing the proof. $\square$

We close this section with the proof of Lemma 8.

Proof of Lemma 8 Consider any constraint-path P which can be generated from $\mathcal{C}^*$. We will begin by showing that there is a constraint-path Q which can be generated from $\mathcal{C}^*$ such that (i) the conjunction PQ does not permit (x, x) for any $x \in \mathcal{D}$; and (ii) Q has length at most q, for some constant $q = q(\mathcal{C}^*)$. Note that (i) implies that identifying the endpoints of PQ will produce an unsatisfiable cyclic CSP.

Let σ be the permutation on $\{1, ..., t\}$ corresponding to P. Let $P(\sigma)$ be the shortest constraint-path inducing σ that can be generated from $\mathcal{C}^*$. Let r be the product of the lengths of the cycles of σ. $P(\sigma)^r$, the concatenation of r copies of $P(\sigma)$, induces the identity permutation, and so $P(\sigma)^{r-1}$ induces σ^{-1}. Recall from the proof of Lemma 19 that $\mathcal{C}^*$ generates a constraint C which induces a permutation with no fixed points; let T be the corresponding constraint-path. Setting $Q = P(\sigma)^{r-1}T$, we see that PQ induces that permutation with no fixed points. Thus, PQ does not permit (x, x) for any x.

Finally, note that Q is defined only in terms of $\mathcal{C}^*$ and σ, and so we can set q to be the length of the longest of the at most $t!$ choices for Q.

For any i, let R_i be the set of cyclic CSP's with lengths in $\{iq+1, ..., (i+1)q\}$ that can formed from $\mathcal{C}^*$. Consider a random member of R_i chosen as follows: First choose a uniform number l from $\{iq+1, ..., (i+1)q\}$; then choose a cyclic CSP of length l whose constraints are chosen independently with distribution $\mathcal{P}$.

Claim: There is some $\zeta > 0$ such that for any i, this CSP will be unsatisfiable with probability at least ζ.

Proof: Before exposing l, we can expose the choice of the first iq consecutive constraints, and let P be the resultant constraint-path. We have argued that there is some Q of length at most q such that if (a) we choose $l = iq + |Q|$ and (b) we choose the remaining constraints to form Q then the resultant CSP will be unsatisfiable. Choice (a) occurs with probability $1/q$, and the probability of choice (b) is a function only of Q and $\mathcal{P}$. Neither of these depends on i, and so there is a $\zeta > 0$ such that, for any choice of P, the probability that the cyclic CSP is unsatisfiable is at least ζ. This proves our claim. □

For each $i = O(1)$, the expected number of unsatisfiable cyclic sub-CSP's of $CSP_{n,M=cn}(\mathcal{P})$ which are in R_i is the sum over $iq < j < (i+1)q$ of the expected number of j-cycles in the underlying graph, times the probability that placing constraints with distribution $\mathcal{P}$ on a j-cycle creates an unsatisfiable CSP. The expected number of j-cycles is well-known (and easily computed) to be $(2c)^j/2j$. Since this is decreasing with j for $c < \frac{1}{2}$, our Claim implies that this expected number is at least $\zeta \times (2c)^{(i+1)q}/2(i+1)q$. Thus, for any fixed i_0 the total expected number of unsatisfiable cyclic sub-CSP's in $CSP_{n,M=cn}(\mathcal{P})$ is at least:

$$\sum_{i=0}^{i_0} \frac{\zeta(2c)^{(i+1)q}}{2(i+1)q}.$$

For $c = 1$, this sum tends to infinity with i_0, and so for any K, we can choose i_0 and $c < \frac{1}{2}$ so that this sum is greater than K. Using very standard facts about the distribution of cycles in the random graph $G_{n,M=cn}$ when $c < \frac{1}{2}$ (see, eg. Theorem 3.19 and Remark 3.20 of [28]), it is easy to show that the distribution of the number of unsatisfiable cyclic sub-CSP's is asymptotic to a Poisson variable. Thus, the probability that $CSP_{n,M=cn}(\mathcal{P})$ is satisfiable is at most $\mathrm{e}^{-K} + o(1)$, which is less than ϵ for K sufficiently large. □

6 Proofs of the main theorems

Proof of Theorem 2

Case 1: supp($\mathcal{P}$) *is connected.*

Since supp($\mathcal{P}$) is connected, its only component is very-well-behaved. Thus, we are in case (a) of Theorem 2, and so we want to show there is some $c > \frac{1}{2}$ such that $CSP_{n,M=cn}(\mathcal{P})$ is a.s. satisfiable.

Set $\mathcal{C} = \text{supp}(\mathcal{P})$ and recall from the remark in Section 3 that we can assume $\mathcal{C}$ to be reflective. First we prove that $\mathcal{C}$ can form a semi-null constraint-path. Assume otherwise, and consider the set $\mathcal{C}^*$ guaranteed to exist by Lemma 17. Since $\mathcal{C}$ is connected, and every member of $\mathcal{C}$ is a subconstraint of some member of $\mathcal{C}^*$, we have that $\mathcal{C}^*$ is connected. Therefore, by Lemma 19, $\mathcal{C}^*$ can form an unsatisfiable cyclic CSP, f. Consider the cyclic CSP f' obtained by replacing each constraint C^* of f by a constraint $C \in \mathcal{C}$ such that $C \subseteq C^*$. (Such a constraint is guaranteed to exist by condition (ii) of Lemma 17.) Any satisfying solution of f' would be a satisfying solution to f and so f' is unsatisfiable. This contradicts the hypothesis that $\text{supp}(\mathcal{P})$ is very-well-behaved.

Thus, $\mathcal{C} = \text{supp}(\mathcal{P})$ can form a semi-null constraint-path and so Lemma 6(a) implies that there is some $\epsilon > 0$ such that for all $c \leq \frac{1}{2} + \epsilon$, $CSP_{n,M=cn}(\mathcal{P})$ is a.s. satisfiable.

This completes the proof of Theorem 2 for the case that $\text{supp}(\mathcal{P})$ is connected.

Case 2: $\text{supp}(\mathcal{P})$ *is not connected*

Again, set $\mathcal{C} = \text{supp}(\mathcal{P})$ and recall that we can assume $\mathcal{C}$ to be reflective. Suppose that $\mathcal{C} = \text{supp}(\mathcal{C})$ is disconnected, with components $\mathcal{C}_1, ..., \mathcal{C}_t$, which have value sets $V_1, ..., V_t$.

Note that if f is a CSP whose constraints are all from $\mathcal{C}$, then in any satisfying assignment of f, each component of the constraint graph will have all of its values taken from some V_i. (We can use different V_i's for different components of the constraint graph).

Theorem 2(a):

Suppose $\mathcal{C}_i$ is a very-well-behaved component. Since $\mathcal{C}$ is reflective, so is $\mathcal{C}_i$. Furthermore, it is connected. Therefore, Case 1 applies to $\mathcal{C}_i$ and so there is some $c > \frac{1}{2}$, such that a.s. $CSP_{n,M=cn}(\mathcal{P})$ can be satisfied using only values from V_i.

Theorem 2(b):

Consider any $c = \frac{1}{2} + \epsilon$ for some $\epsilon > 0$. Let F be the sub-CSP induced by the giant component of the underlying graph of $CSP_{n,M=cn}(\mathcal{P})$. We will prove that a.s. F cannot be satisfied by the values of any component of $\mathcal{C}$.

Consider any component $\mathcal{C}_i$ which has no good values. Then by Lemma 1, there is a CSP f_i formed from $\mathcal{C}_i$ whose underlying graph is a tree. It is straightforward to prove that a.s. f_i appears as a sub-CSP of F. Thus a.s. F cannot be satisfied using the values of $\mathcal{C}_i$.

Consider any component $\mathcal{C}_i$ which is a subpermutation set. Let $c' = \frac{1}{2} + \epsilon/2$. We can select $CSP_{n,M=cn}(\mathcal{P})$ by first selecting $CSP_{n,M=c'n}(\mathcal{P})$ and then selecting $\epsilon n/2$ additional random constraints to add to it. There is some $\delta = \delta(\epsilon)$ such that a.s. the underlying graph of $CSP_{n,M=c'n}(\mathcal{P})$ has a giant component of size at least δn; let F' be the sub-CSP induced by that component.

Given any satisfying assignment of F', using the values of $\mathcal{C}_i$, we label the variables of F' with values from $1, ..., t$ as follows: Suppose that the underlying partition of $\mathcal{C}_i$ is $A_1, ..., A_t$ (and so $t \geq 2$). If variable v is assigned a value in A_j, then the label of v is j. Consider a particular variable u and some $1 \leq j \leq t$ and consider all satisfying assignments for which u receives label j. Since every constraint in $\mathcal{C}_i$ is contained by some expanded permutation with partition $A_1, ..., A_t$, the labels of all the other variables in these assignments are forced by the fact that u receives label j. Thus there is some label j' and set X of variables of size at least $\delta n/t$ such that in any satisfying assignment where u has label j, every variable of X has label j'. Since $\mathcal{C}_i$ is connected, there is at least one constraint $C \in \mathcal{C}_i$ which forbids every pair (x, y) with $x, y \in A_{j'}$. Thus, when we add the $\epsilon n/2$ additional random constraints, each one has a $(\delta/t)^2\mathcal{P}(C)$ chance of forbidding some pair of variables in X from both receiving values from $A_{j'}$. Thus, the probability that $CSP_{n,M=cn}(\mathcal{P})$ has a satisfying assignment in which u has a value from A_j is at most $(1-(\delta/t)^2\mathcal{P}(C))^{\epsilon n/2}$. Even after multiplying by t for the number of choices for j, this is $o(1)$. Therefore, a.s. $CSP_{n,M=cn}(\mathcal{P})$ does not have a satisfying assignment using only values from $\mathcal{C}_i$

Theorem 2(c):

We start with the existence of γ_1. Consider any $c > \frac{1}{2}$.

For each i, $\mathcal{C}_i$ is not very-well-behaved, i.e. it does not have all 3 of Properties 1, 2 and 3 from Section 3. Since $\mathcal{C}$ has Property 1, then so does each $\mathcal{C}_i$. Thus each $\mathcal{C}_i$ fails either Property 2 or 3, and so by Lemma 1, there is some CSP f_i formed from $\mathcal{C}$ which is either a tree CSP or a unicyclic CSP and such that the projection of f_i to V_i is unsatisfiable. It is straightforward to show that with probability at least $\gamma + o(1)$, for some constant $\gamma = \gamma(\mathcal{P}, c) > 0$, f_i is a sub-CSP of the giant component of $CSP_{n,M=cn}(\mathcal{P})$ for every i. (We omit the simple, but tedious, proof.) As noted above, in any satisfying assignment, there must be some i such that all the variables in the giant component receive a value from V_i. But this is not possible if every f_i is in the giant component. Therefore, $CSP_{n,M=cn}(\mathcal{P})$ is satisfiable with probability at most $\gamma_1 + o(1)$ where $\gamma_1 = 1 - \gamma$.

Now we turn to the existence of γ_2. By hypothesis, there is a component $\mathcal{C}_i$ of $\mathcal{C}$ which has at least one good value and which is not a subpermutation set. By Lemma 17, $\mathcal{C}_i$ can form a semi-null constraint-path (to see this note that if there is some $\mathcal{C}^*$ satisfying (i) and (iii) of Lemma 17 then some subset of $\mathcal{C}^*$ satisfies (i), (ii) and (iii) of Lemma 17). Since supp$(\mathcal{P})$ is very-well-behaved, every tree and unicyclic CSP formed from supp$(\mathcal{P})$ is satisfiable. Furthermore, a.s. every component but the giant component of $CSP_{n,M=cn}(\mathcal{P})$ either is a tree or is unicyclic. Therefore, we can take $\gamma_2(\mathcal{P}, c)$ to be ξ from Lemma 7.

This completes the proof of Theorem 2. □

Proof of Theorem 4 For $c < \frac{1}{2}$, the underlying graph is a.s. such that every component has at least one cycle. Thus, since $\mathcal{C}$ is well-behaved, a.s. the random CSP is unsatisfiable iff it has an unsatisfiable cyclic sub-CSP.

Consider any $\zeta > 0$. A standard first moment calculation shows that for a sufficiently large constant t, the probability that the underlying graph has a cycle of length greater than t is at most ζ. Indeed, in the graph $G_{n,p=2c/n}$ the expected number of cycles of length greater than t is at most:

$$\sum_{i>t} \binom{n}{i} \frac{(i-1)!}{2} p^i = \sum_{i>t} (2i)^{-1} (2c)^i < \zeta$$

for large t since $2c < 1$. For such t, we define X_t to be the set of unsatisfiable cyclic CSP's of length at most t that can be formed from supp$(\mathcal{P})$ and for each $H \in X_t$ we set $\mu_H = \mu_H(c)$ to be the limit as $n \to \infty$ of the expected number of occurrences of H in $CSP_{n,M=cn}(\mathcal{P})$; it is straightforward to verify that $0 < \mu_H < 1$. Set $u_t = u_t(c) = \sum_{H \in X_t} \mu_H$. Standard techniques (eg. those in Section 3.3 of [28]) show that the probability that $CSP_{n,M=cn}(\mathcal{P})$ includes no members of X_t is $e^{-u_t} + o(1)$. Thus, the probability that $CSP_{n,M=cn}(\mathcal{P})$ is satisfiable is between $e^{-u_t} + o(1)$ and $e^{-u_t} + \zeta + o(1)$. By letting ζ shrink and t grow, this implies the probability that $CSP_{n,M=cn}(\mathcal{P})$ is satisfiable is $\alpha(c) + o(1)$ where $\alpha(c) = \exp(-\lim_{t\to\infty} u_t)$. (That limit is easily seen to converge since all terms are positive and the total weight of its tail tends to 0.) Finally, it is easy to see that each μ_H is continuous and monotonically increasing and that $\alpha(c)$ is continuous and monotonically decreasing.

Now we turn our attention to the latter portion of the theorem.

The first statement of part (a) follows in the same way as the proof of Theorem 2(b). The second statement of part (a) follows immediately from Corollary 9.

For part (b): since the hypothesis of part (a) doesn't hold, there is at least one component $\mathcal{C}_i$ of supp($\mathcal{P}$) that has at least one good value and is not subsumed by an expanded permutation with at least 2 parts. So Lemma 17 implies that $\mathcal{C}_i$ can form a semi-null constraint-path, and so Lemmas 6 and 7 apply to $\mathcal{C}_i$.

By Lemma 6, there is $\epsilon > 0$ and $\gamma > 0$ such that for $c \leq \frac{1}{2} + \epsilon$ the probability that $CSP_{n,M=cn}(\mathcal{P})$ is satisfiable is at least $\gamma + o(1)$. Therefore, $\alpha(\frac{1}{2}) \geq \gamma$. Note that for $c > \frac{1}{2}$, the probability that $CSP_{n,M=cn}(\mathcal{P})$ is unsatisfiable is bounded from above by the probability that the giant component cannot be satisfied using only values from $\mathcal{C}_i$ plus the probability that one of the other components is unsatisfiable. By Lemma 7, for c sufficiently close to $\frac{1}{2}$ this is arbitrarily close to the latter probability, which is arbitrarily close to $\alpha(\frac{1}{2})$. □

7 The case $k \geq 3$

We close this paper with the case $k \geq 3$.

The definitions from Section 3 of *reflective*, *well-behaved* and *very-well-behaved* extend to $k \geq 3$ in the obvious way.

To extend the definition of a bad value we say that a value is *j-bad* if there is some canonical variable X_i, and constraint $C \in \mathcal{C}$ such that C implies that if $X_i = \delta$ then *at least one* other canonical variable must be assigned a j'-bad value for some $j' < j$; the extension of the rest of the definition is in the obvious way.

A *tree CSP*, *cyclic CSP*, *unicyclic CSP* is one whose constraint hypergraph is respectively a hypertree, a hypercycle, a unicyclic connected hypergraph. Formally, in the k-uniform case, we define: a hypercycle is a hypergraph with hyperedges $e_1, ..., e_m$, m vertices of degree 2, and $m(k-2)$ vertices of degree 1; each e_i contains a degree 2 vertex which it shares with e_{i-1}, a different degree 2 vertex which it shares with e_{i+1} (addition and subtraction are mod m), and $k-2$ degree 1 vertices (which, of course, lie in no other edges). Note that two edges which share two vertices form a hypercycle. A hypertree is a connected hypergraph which contains no subset of hyperedges that form a hypercycle and in which no two hyperedges share at least 2 vertices. A *unicyclic* connected hypergraph is one which has exactly one subset of hyperedges that form a hypercycle and in which no two hyperedges share at least 3 vertices.

The definitions of Properties 1, 2, 3 extend in the obvious way. As does Lemma 1, which was also proved in [36]:

Lemma 20 *(a) $\mathcal{C}$ can form an unsatisfiable hypertree CSP iff it does not satisfy Property 2.*

(b) If $\mathcal{C}$ satisfies Property 2, then it can form an unsatisfiable unicyclic CSP iff it does not satisfy Property 3.

Let $H_{n,M}^{(k)}$ denote the random k-uniform hypergraph on n vertices obtained by choosing M uniformly random hyperedges independently without replacement. A standard result from random hypergraph theory (see eg. [34]) says that if $c > \frac{1}{k(k-1)}$ then a.s. $H_{n,M=cn}^{(k)}$ has a giant component on $\Theta(n)$ vertices and if $c < \frac{1}{k(k-1)}$ then a.s. all the components of $H_{n,M=cn}^{(k)}$ are of size $O(\log n)$ and have a fairly simple structure.

To extend Theorem 2 to the case $k \geq 3$, we need the following new definitions.

A constraint C *permits* $X_1 = u, X_k = v$ if there is at least one k-tuple $(u, \delta_2, ..., \delta_{k-1}, v)$ such that $X_1 = u, X_2 = \delta_2, ..., X_k = v$ satisfies C. $C^{(2)}$ is the constraint on two variables, X_1, X_2 where $C^{(2)}$ permits $X_1 = u, X_2 = v$ iff C permits $X_1 = u, X_k = v$. Given a collection of constraints $\mathcal{C}$ on $X_1, ..., X_k$, we define $\mathcal{C}^{(2)} = \{C^{(2)} : C \in \mathcal{C}\}$. Of course, when $k = 2$, $\mathcal{C}^{(2)} = \mathcal{C}$.

Our full generalization of Theorem 2 to all $k \geq 2$ is:

Theorem 21 *Consider any $d, k \geq 2$ and any distribution $\mathcal{P}$ on $\mathcal{C}^{d,k}$. Set $\mathcal{C} = \mathrm{supp}(\mathcal{P})$, and suppose that $\mathcal{C}^{(2)}$ is very-well-behaved.*

(a) If $\mathcal{C}^{(2)}$ has a very-well-behaved component then there is some $c > \frac{1}{k(k-1)}$ such that $CSP_{n,M=cn}(\mathcal{P})$ is a.s. satisfiable.

(b) Else if every component of $\mathcal{C}^{(2)}$ with at least one good value is a subpermutation set then for every $c > \frac{1}{k(k-1)}$, $CSP_{n,M=cn}(\mathcal{P})$ is a.s. unsatisfiable.

(c) Else then there exists $\epsilon > 0$ such that for every $\frac{1}{k(k-1)} < c < \frac{1}{k(k-1)} + \epsilon$ there exists $\gamma_1 = \gamma_1(\mathcal{P}, c) < 1$ and $\gamma_2 = \gamma_2(\mathcal{P}, c) > 0$ such that

$$\gamma_1 + o(1) \geq \Pr(CSP_{n,M=cn}(\mathcal{P}) \text{is satisfiable}) \geq \gamma_2 + o(1).$$

Furthermore, $\gamma_2(c)$ approaches 0 as c approaches $\frac{1}{k(k-1)}$.

Thus, for each $\frac{1}{k(k-1)} < c \leq \frac{1}{k(k-1)} + \epsilon$, $CSP_{n,M=cn}(\mathcal{P})$ is neither a.s. satisfiable nor a.s. unsatisfiable.

Note that if $\mathcal{C}$ is connected then so is $\mathcal{C}^{(2)}$, and so the only component of $\mathcal{C}^{(2)}$ must be very-well-behaved. Thus our theorem implies:

Corollary 22 *There is no $\mathcal{P}$ with* $\mathrm{supp}(\mathcal{P})$ *connected, such that* $\Pr(CSP_{n,M=cn}(\mathcal{P})$ *is unsatisfiable*$) = o(1)$ *for all $c < \frac{1}{k(k-1)}$ and* $\Pr(CSP_{n,M=cn}(\mathcal{P})$ *is unsatisfiable*$) = \Theta(1)$ *for all $c > \frac{1}{k(k-1)}$.*

Proof of Theorem 21: First we must extend the definition of a semi-null constraint-path to $k \geq 3$. A *constraint-path* of $\mathcal{C}$ with length ℓ is a sequence of distinct variables $v_0, v_1, ..., v_\ell$, and $w_{1,2}, ..., w_{1,k-1}, w_{2,2}, ..., w_{2,k-1}, ...w_{\ell,2}, ..., w_{\ell,k-1}$, with a constraint from $\mathcal{C}$ on each k-tuple $v_{i-1}, w_{i,2}, ..., w_{i,k-1}, v_i$ (where v_{i-1} corresponds to X_1, $w_{i,j}$ corresponds to X_j and v_i corresponds to X_k). We call v_0, v_ℓ the endpoints of the constraint-path and we call $v_1, ..., v_{\ell-1}$ and all $w_{i,j}$ the internal variables. We say that such a constraint-path is *formed* by $\mathcal{C}$. A constraint-path of length ℓ *permits* (x, y) if there is some set of assignments to the internal variables which, along with setting $v_0 = x, v_\ell = y$, does not violate any of the constraints in the path. A constraint-path of length ℓ is *null* if it permits (x, y) for every pair x, y in the domain. It is *semi-null* if it permits (x, y) for every pair of good values x, y in the domain.

Reducing the general case $k \geq 3$ to $k = 2$ requires the following two simple observations:

Proposition 23 *If $\mathcal{C}^{(2)}$ can form a cyclic CSP which cannot be satisfied using good values then so can $\mathcal{C}$.*

Proposition 24 *$\mathcal{C}$ can form a semi-null constraint-path iff $\mathcal{C}^{(2)}$ does.*

The proofs are straightforward after observing that since we can assume $\mathcal{C}$ to be reflective (see the remark in Section 3), it follows that δ is a good value for $\mathcal{C}^{(2)}$ iff it is a good value for $\mathcal{C}$. We omit the details.

Suppose that $\mathcal{C}^{(2)}$ is connected. Since $\mathcal{C}$ is well-behaved, it cannot form a cyclic CSP which cannot be satisfied using good values, and so neither can $\mathcal{C}^{(2)}$ (by Proposition 23). Therefore, by Lemma 7, $\mathcal{C}^{(2)}$ can form a semi-null constraint-path, and thus so can $\mathcal{C}$ (by Proposition 24). A branching process argument nearly identical to the one in the proof of Lemma 6, shows that there exists $\epsilon > 0$ such that for $c = \frac{1}{k(k-1)} + \epsilon$, a.s. $CSP_{n,M=cn}(\mathcal{P})$ is satisfiable.

The case where $\mathcal{C}^{(2)}$ is disconnected follows from the connected case using the same arguments as in section 6. We omit the details as they are straightforward and repetitive.

This completes the proof of Theorem 21. □

Our generalization of Theorem 4 to all $k \geq 2$ is:

Theorem 25 *Consider any $d, k \geq 2$ and any distribution $\mathcal{P}$ on $\mathcal{C}^{d,k}$. Set $\mathcal{C} = \text{supp}(\mathcal{P})$ and suppose that $\mathcal{C}^{(2)}$ is well-behaved but not very-well-behaved. Then there is a continuous, monotonically decreasing function $\alpha : [0, \frac{1}{2}] \to [0, 1]$ with $\alpha(0) = 1$ such that for $0 \leq c < \frac{1}{2}$:*

$$\mathbf{Pr}(CSP_{n,M=cn}(\mathcal{P})\text{is satisfiable}) = \alpha(c) + o(1).$$

Furthermore, for the case where $\mathcal{C}^{(2)}$ is connected we have:

(a) If each component of $\mathcal{C}^{(2)}$ with at least one good value is a subpermutation set then for every $c \geq \frac{1}{2}$, $CSP_{n,M=cn}(\mathcal{P})$ is a.s. unsatisfiable. If, in addition, $\mathcal{C}^{(2)}$ is connected then $\alpha(\frac{1}{2}) = 0$.

(b) Else:

(i) $\alpha(\frac{1}{2}) > 0$; and

(ii) There exists $\epsilon > 0$ such that for every $\frac{1}{2} < c \leq \frac{1}{2} + \epsilon$ there exists $\gamma_2 = \gamma_2(c) > 0$ such that

$$\Pr(CSP_{n,M=cn}(\mathcal{P})\text{is satisfiable}) \geq \gamma_2 + o(1).$$

Furthermore, $\gamma_2(c)$ approaches $\alpha(\frac{1}{2})$ as c approaches $\frac{1}{2}$ from above.

Thus, for each $\frac{1}{2} < c \leq \frac{1}{2} + \epsilon$, $CSP_{n,M=cn}(\mathcal{P})$ is neither a.s. satisfiable nor a.s. unsatisfiable.

Proof The proof follows just like the proof of Theorem 4 after extending the concepts and arguments to hypergraphs, just as we did for the proof of Theorem 21. It is straightforward to extend the arguments of Lemmas 6, 7, 8 and Corollary 9. We omit the details. □

8 Conclusion and Open Problems

We have established which members of our family of models are such that they remain a.s. unsatisfiable beyond the appearance of a giant component in the underlying CSP. This determines which models exhibit a satisfiability threshold that is in some sense non-trivial. This is just a beginning; there are still many properties that we would like to know about such thresholds, beyond this non-triviality.

The most obvious remaining question is: which of these models exhibit a sharp threshold of satisfiability? Using the traditional notion of a sharp threshold, this asks: for what distributions $\mathcal{P}$ does there exist a constant c^* such that for $c < c^*$, $CSP_{n,M=cn}(\mathcal{P})$ is a.s. satisfiable, while for $c > c^*$, $CSP_{n,M=cn}(\mathcal{P})$ is a.s. unsatisfiable. This is the driving question behind most of the research surrounding individual random models from our family (see eg. [4, 8, 33] etc.) Friedgut[23] introduced a weaker notion of sharpness where the constant c^* is replaced by a function $c^*(n)$. Most of the more heavily-pursued models from this family, most notably random 3-SAT[23], are known to have sharp thresholds in the weaker case, and are *conjectured but not known* to have sharp thresholds in the stronger sense. Until this problem is settled for these important special cases, it is overly ambitious to try to answer our question for the stronger notion of a sharp threshold.

The author's opinion is that the most important open line of pursuit regarding the class of models studied in this paper is the question of which exhibit satisfiability thresholds that are sharp in the sense of Friedgut[23]. Istrate[26] and independently Creignou and Daudé[17] solved this for the case where $k = 2$, proving that if supp$(\mathcal{P})$ is very-well-behaved then $CSP(\mathcal{P})$ has a sharp threshold. (Actually, in each paper they only proved it for a subset of the models that have $k = 2$, but it is straightforward to extend their proofs to the case of all models with $k = 2$ by using a sufficiently strong version of Freidgut's Theorem, eg. the one used in [27].) The analogous result does not hold for $k \geq 3$ (see [36] for a counterexample). Some simple partial results for the more general problem appear in [36], and more substantial partial results appear in [27].

Also in [27], an example of a particular $\mathcal{P}$ was presented such that determining whether $CSP(\mathcal{P})$ has a sharp satisfiability threshold requires determining the location of the 3-colourability threshold. This shows that characterizing those values of $\mathcal{P}$ for which $CSP(\mathcal{P})$ has a sharp satisfiability threshold is very hard indeed. In the example given in [27], supp$(\mathcal{P})$ is disconnected. Perhaps it would be better to just try to determine those $\mathcal{P}$ with connected supp$(\mathcal{P})$ for which $CSP(\mathcal{P})$ has a sharp satisfiability threshold. Or, maybe instead of obtaining a through characterization, one could provide something slightly weaker in the form of a description of exactly what can cause coarse thresholds.

Here is another indication that the problem of determining which models have sharp satisfiability thresholds is much more insidious than the one solved in the present paper: The aforementioned examples from [27, 36] show that there are pairs of distributions $\mathcal{P}_1, \mathcal{P}_2$ with supp$(\mathcal{P}_1)$ = supp$(\mathcal{P}_2)$ and for which $CSP(\mathcal{P}_1)$ has a sharp threshold but $CSP(\mathcal{P}_2)$ does not. In other words, it will not suffice to focus only on properties of $\mathcal{C}$ = supp$(\mathcal{P})$, as it did in this paper; we have to look at the actual probability distribution.

Another important line of pursuit is to determine which models from this family exhibit high resolution complexity. A seminal paper by Chvatál and Szemerédi[15] showed that for every constant $c > 0$, a random instance of 3-SAT with cn clauses a.s. has no sub-exponential sized resolution proof of unsatisfiability. This implies that every resolution based algorithm (for example, all the standard implementations of Davis-Putnam) will a.s. take exponential time on an unsatisfiable instance. This result has been extended in many directions (see eg. [11]) Mitchell[35] laid down the groundwork to extend this line of pursuit to random constraint satisfaction problems from the family studied in the present paper. Molloy and Salavatipour[37] determined precisely which members of a large natural subfamily of models a.s. have exponentially high resolution complexity. As with the previous question, it does not suffice here to focus on properties of $\mathcal{C} = \mathrm{supp}(\mathcal{P})$. Furthermore, there are distributions $\mathcal{P}$ such that for some values of c, $CSP_{n,M=cn}(\mathcal{P})$ a.s. has exponentially high resolution complexity while for other values of c, it a.s. has polynomial resolution complexity. So again, this problem appears to be more complicated than the one solved in the present paper.

More recently, exponential lower bounds on unsatisfiablity proofs from various other proof systems have been established for random 3-SAT. It would be natural to also extend these studies to our more general family of models.

An important question arising from the point of view of the statistical physics community, is to determine which models from this family exhibit a first order (i.e. discontinuous) phase transition and which exhibit a second order (i.e. discontinuous) phase transition. See, eg [38] for definitions and a more detailed introduction to the problem. See [3] for an example of a rigorous analysis of the phase transition order for two specific models.

There are many other questions that can be asked. For example, we mentioned in the introduction that the threshold of satisfiability of random 2-SAT coincides with the appearance of the giant strongly connected component of a natural underlying random digraph. Perhaps there is an interesting general phenomenon here, and one might ask what other models from our family behave in a similar manner.

Acknowledgements

The author is grateful to Dimitris Achlioptas, Henry Cohn and Jeong Han Kim for many fruitful discussions, and to two anonymous referees for suggesting many improvements.

References

[1] D. Achlioptas. *Setting two variables at a time yields a new lower bound for random 3-SAT.* Proceedings of STOC 00, 28-37.

[2] D. Achlioptas, P. Beame and M. Molloy. *A sharp threshold in proof complexity.* Proceedings of STOC 2001, 337 - 346.

[3] D. Achlioptas, A. Chtcherba, G. Istrate and C. Moore. *The phase transition in NAESAT and 1-in-k SAT.* Proceedings of SODA 01, 721 - 722.

[4] D. Achlioptas and E. Friedgut. *A threshold for k-colourability.* Random Structures and Algorithms **14** (1999), 63 - 70.

[5] D. Achlioptas, L. Kirousis, E. Kranakis, D. Krizanc, M. Molloy, and Y. Stamatiou. *Random constraint satisfaction: a more accurate picture.* Constraints **6**, 329 - 324 (2001). Conference version in Proceedings of CP 97, 107 - 120.

[6] D. Achlioptas and M. Molloy. *Analysis of a list-coloring algorithm on a random graph.* Proceedings of FOCS 97, 204-212.

[7] D. Achlioptas and M. Molloy. *Almost all graphs with 2.522n edges are not 3-Colorable.* Electronic Journal of Combinatorics, 6 (1), (1999), R29

[8] D. Achlioptas and C. Moore. *The asymptotic order of the k-SAT threshold* Proceedings of FOCS 2002.

[9] D. Achlioptas and C. Moore. *Almost all graphs of degree 4 are 3-colorable.* Proceedings of STOC 2002.

[10] D. Achlioptas and G.B Sorkin. *Optimal myopic algorithms for random 3-SAT* Proceedings of FOCS 00, 590-600.

[11] P. Beame, R. Karp, T. Pitassi and M. Saks. *The efficiency of resolution and Davis-Putnam procedures.* SIAM J. Comp. **31** (2002) 1048 - 1075.

[12] B.Bollobás. *Random Graphs.* Academic Press, London, 1985.

[13] B. Bollobás, C. Borgs, J. T. Chayes, J. H. Kim, and D. B. Wilson. *The scaling window of the 2-SAT transition.* Random Structures and Algorithm **18** (2001), 201 - 256.

[14] V. Chvátal and B. Reed. Mick gets some (the odds are on his side). In *Proceedings 33rd Annual Symposium on Foundations of Computer Science*, pages 620–627, Pittsburgh, PA, October 1992. IEEE.

[15] V. Chvatal and E. Szemeredi. *Many hard examples for resolution.* Journal of the ACM **35** (1988) 759 - 768.

[16] N. Creignou and H. Daude. *Generalized satisfiability problems: minimal elements and phase transitions.* Th. Comp. Sci. **302** (2003), 417 - 430.

[17] N. Creignou and H. Daude. *Combinatorial sharpness criterion and phase transition classification for random CSP's.* Information and Computation **190** (2004), 220 - 238.

[18] O. Dubois and J. Mandler. *On the non-3-colourability of random graphs.* submitted.

[19] O. Dubois, Y. Boufkhad and J. Mandler. *Typical random 3-SAT formulae and the satisfiability threshold.* Proceedings of 11th Symposium on Discrete Algorithms (SODA), 126 127 (2000).

[20] P. Erdos and A. Renyi. *On the evolution of random graphs.* Mat Kutato Int Kozl **5** (1960), 17 - 60.

[21] W. Fernandex de la Vega. *On random 2-SAT.* manuscript (1992).

[22] N. Fountoulakis and C. McDiarmid. *Upper bounds on the non-colourability threshold of random graphs.* submitted.

[23] E. Friedgut and an appendix by J. Bourgain. *Sharp thresholds of graph properties and the k-SAT problem.* J. Am. Math. Soc. **12** (1999), 1017 - 1054.

[24] I. Gent, E. MacIntyre, P. Prosser, B. Smith and T. Walsh. *Random constraint satisfaction: flaws and structure.* Constraints **6**, 345 - 372 (2001).

[25] A. Goerdt. *A threshold for unsatisfiability.* J. Computer and System Sciences **53** (1996), 469–486.

[26] G. Istrate. *Threshold properties of random constraint satisfaction problems.* Disc. Appl. Math. (to appear).

[27] H. Hatami and M. Molloy. *Sharp thresholds for constraint satisfaction problems and homomorphisms.* Submitted.

[28] S. Janson, T. Łuczak and A. Ruciński. *Random Graphs.* Wiley, New York (2000).

[29] S. Janson, Y. Stamatiou and M. Vanvakari. *Bounding the unsatisfiability threshold of random 3-SAT.* Random Structures and Algorithms **17** (2000), 103 - 116.

[30] R. Karp. *The transitive closure of a random digraph.* Random Structures and Algorithms **1** (1990), 73 - 94.

[31] A. Kaporis, L. Kirousis and E. Lalas. *The probabilistic analysis of a greedy satisfiability algorithm.* Proceedings of ESA 2002, 574 - 585

[32] A. Kaporis, L. Kirousis and Y. Stamatiou. *A note on the non-colourability threshold of a random graph.* Electronic J. Comb, **7** R29 (2000)

[33] L. Kirousis, E. Kranakis, D. Krizanc and Y. Stamatiou. *Approximating the unsatisfiability threshold of random formulas.* Random Structures and Algorithms **12** (1998), 253 - 269.

[34] T. Łuczak and M. Karonski. *The phase transition in a random hypergraph.* J. Comput. Appl. Math. 142 (2002) 125-135.

[35] D. Mitchell. *Resolution complexity of random constraints.* Proceedings of Principles and Practices of Constraint Programming - CP 2002.

[36] M. Molloy, *Models for Random Constraint Satisfaction Problems.* Proceedings of STOC 2002, 209 - 217. Longer version to appear in SIAM J. Computing.

[37] M. Molloy and M. Salavatipour. *The resolution complexity of random constraint satisfaction problems.* (submitted). Conference version in Proceedings of FOCS 2003.

[38] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky. *Determining computational complexity from characteristic 'phase transitions'.* Nature, **400(8)** (1999), 133–137.

[39] D.S. Passman, *Permutation Groups.* Benjamin, New York (1968).