Resolution Complexity of Random Constraints

David G. Mitchell

Simon Fraser University
Burnaby, Canada
{mitchell@cs.sfu.ca}

Abstract. Random instances are widely used as benchmarks in evaluating algorithms for finite-domain constraint satisfaction problems (CSPs). We present an analysis that shows why deciding satisfiability of instances from some distributions is challenging for current complete methods. For a typical random CSP model, we show that when constraints are not too tight almost all unsatisfiable instances have a structural property which guarantees that unsatisfiability proofs in a certain resolution-like system must be of exponential size. This proof system can efficiently simulate the reasoning of a large class of CSP algorithms which will thus have exponential running time on these instances.

1 Introduction

Randomly generated CSP instances are widely used as benchmark instances in the empirical evaluation of algorithms, and have been since the earliest work in the area. They are also used in average case analyses of algorithms, and as combinatorial objects of independent interest. A series of papers beginning with [7] and [20] established that, while random instances are often easy, with suitable choice of generation model and parameters, instances can be generated which are challenging for current complete algorithms. Prosser[23] studied instances from very natural CSP models, which have been further studied in many papers.

Our goal here is to show that many such instances are indeed hard for a large class of algorithms, and to some extent to give an explanation of why. Our approach is inspired by an important result in propositional proof complexity, namely that randomly generated k-CNF formulas with linearly many clauses almost surely have no resolution refutations of less than exponential size [9]. Backtrack-based algorithms for SAT (e.g., DPLL algorithms) can be viewed as constructing resolution refutations for unsatisfiable formulas, so these algorithms will have exponential running time on unsatisfiable random formulas. This suggests that analogous random CSP instances might be similarly hard for backtracking-based CSP algorithms.

To capture as wide a class of algorithms as possible, we take the following two step approach. One step is to show that for some suitable resolution-like proof system for CSPs, random instances require exponential size refutations. The second is to show that the reasoning used in popular CSP algorithms can be efficiently simulated by derivations in this proof system. In this paper we restrict our attention to the first step. The interested reader will find some preliminary work on the second step in [17] and a more complete treatment in [18, 19].

1.1 Resolution and Constraint Satisfaction

For a CSP refutation system we employ a natural transformation from CSP to SAT, and regard the (propositional) resolution refutations of the resulting *associated CNF formula* to be (CSP) resolution refutations of the CSP instance. We call this system **C-RES**. The original motivation for its study was de Kleer's paper [10] showing a relationship between CSP local-consistency methods and certain resolution derivations in this CNF encoding. Another notion of resolution for CSP was proposed by Baker [4], and studied in [15, 18]. That system is very natural, but is weaker and cannot account for some basic algorithms [18, 19]

The most-used algorithms in the CSP literature can be viewed as implicitly constructing **C-RES** refutations as they execute. In [18, 19] we call an algorithm ***C-RES** bounded* if its number of steps to execute on an instance $\mathcal{I}$ is greater than the size of the smallest **C-RES** refutation $\mathcal{I}$. The set of **C-RES** bounded algorithms includes all those that can be described in terms of (common variants of) the following methods: backtracking search, including variable and value ordering heuristics; conflict-directed backjumping [22]; forward checking and k-consistency filtering [13, 16]; nogood recording (a.k.a. "learning") [11]. This includes most standard techniques from the CSP literature, for example all algorithms described in [12, 22]. Simulations work roughly as follows. When backtracking returns from a recursive call, it can be viewed as returning a derived clause that rules out the assignment made for that call. When a node is exhausted, several such clauses are combined to derive a clause for the level above. Consistency filtering methods derive new clauses as soon as a local inconsistency is recognized. Backjumping observes which variables occur in derived clauses, and nogood recording caches some derived clauses.

1.2 The Sub-critical Expansion Method

Our method is a generalization to CSPs of existing methods for propositional resolution lower bounds. See, e.g., [5] and [6]. We define a measure $\mathcal{E}(I)$ of a CSP instance $\mathcal{I}$ called the *sub-critical expansion*, and show that instances with large $\mathcal{E}(I)$ require large **C-RES** refutations. We then show that random CSP instances, within certain parameter ranges, almost surely have large $\mathcal{E}(I)$.

The intuition is roughly this. The critical size $\mathcal{C}(\mathcal{I})$ is the smallest number of variables in any unsatisfiable sub-instance of $\mathcal{I}$. All "sub-critical" sub-instances are satisfiable. Any **C-RES** refutation of $\mathcal{I}$ must be contain some sub-derivations all of whose input clauses come some sub-critical sub-instance. $\mathcal{E}(\mathcal{I})$ is a measure of how many boundary variables such a sub-instance has. Variable x is a boundary variable if any assignment that satisfies its neighbor variables can be consistently extended to x. If there are many of these, the clause derived from this sub-instance must be large, so in a sense $\mathcal{E}(\mathcal{I})$ is a measure of how much information some clause must carry from a sub-derivation to a refutation of $\mathcal{I}$.

Finally, we can apply a result from [6] relating clause size and refutation size to show that the refutation itself must be large.

Sub-critical expansion is an important structural measure of instances since no **C-RES** refutation can be of size less than $2^{\Omega(\mathcal{E}(\mathbf{I})^2/n)}$. To show that random CSPs have large $\mathcal{E}(I)$, we need to show that they have no small unsatisfiable sub-instances, and that sub-critical sub-instances within a certain size range have large boundaries. Proofs of both of these properties are based on the fact that sub-graphs of random hypergraphs have quite uniform edge density.

1.3 C-RES complexity of Random CSPs

We consider a class of random CSP models typical of those used in experiments, denoting by $\mathsf{CSP}^{d,k,t}_{n,\Delta}$ the random instance $\mathcal{I}$ with domain $[d]$ and constraint arity k, generated as follows. Select $\lfloor \Delta n \rfloor$ constraints uniformly at random from those constraints with tightness t. Tightness, here, is the number of k-tuples of values the constraint excludes.

We fix all parameters except n, and study $\mathsf{CSP}^{d,k,t}_{n,\Delta}$ as $n \to \infty$. A property holds *almost surely* (a.s.) if it holds with probability tending to 1 as $n \to \infty$. We say $\mathsf{CSP}^{d,k,t}_{n,\Delta}$ exhibits a *phase transition* if there are constants c_1,c_2 such that instances are almost surely satisfiable if $\Delta < c_1$ and a.s. unsatisfiable if $\Delta > c_2$. Models with phase transitions are interesting because empirically challenging instances with a probability of being satisfiable of roughly $1/2$ are often associated with densities at or near c_1, c_2.

We first dispense with some algorithmically easy cases. It is shown in [2], that if $t \geq d^{k-1}$ then $\mathsf{CSP}^{d,k,t}_{n,\Delta}$ is almost surely unsatisfiable, independent of Δ (for any $d, k > 1$), so there is no phase transition. This is because there is a.s. an unsatisfiable sub-instance of size $d(k-1)+1$, from which a **C-RES** refutation of size $O(1)$ can be obtained. These can be found efficiently by standard methods to check for $d(k-1)+1$-consistency. CSPs with $d < 2$ or $k < 2$ are trivial. Results in [8] about random 2-SAT, plus simulation results in [18], show that $\mathsf{CSP}^{2,2,1}_{n,\Delta}$ exhibits a transition and a.s. has a **C-RES** refutation of size $O(\log n)$. More generally, all cases with $d = k = 2$ can be solved in polynomial time, for example by transformation to 2-SAT, and it is not hard to check that all instances have **C-RES** refutations of size at most $O(m^2)$, where m is the number of constraints.

We are left to consider the case when $d, k \geq 2$, $d + k > 4$ and $d^{k-1} > t > 0$. For these values of d,k, the CSP problem is **NP**-complete, and it follows from [21, 14] that when $t <= d^{k-1}$ $\mathsf{CSP}^{d,k,t}_{n,\Delta}$ does exhibit a phase transition (and certainly $c_2 < d^{k(t+2)} \ln d$). So we might expect these problems to be harder for resolution. Since we are studying refutations, our results here will only apply in the case $\Delta > c_2$, when most instances are unsatisfiable. We will leave this assumption implicit in this section. Notice though, that a backtrack-based algorithm can only can only take a long time if it generates an unsatisfiable instance that takes a long time to refute, so this work is not irrelevant in the satisfiable case. Indeed, recent results in [1] suggest that similar, though weaker, results may be obtained for such densities below c_2.

If we fix $d = 2$, and $t = 1$, we obtain a random CSP equivalent to random k-SAT. The lower bounds for these cases, when $k \geq 3$, can be obtained from the random k-SAT lower bounds of [9], plus simulation arguments given in [18]. So the main question here is to what extent we can extend the range of parameters for which such lower bounds hold.

Our main result is that if $d, k > 1$, $d + k > 4$, and $(d-1)(k-1) > t > 0$ then almost surely $\mathbf{C\text{-}RES}(\mathsf{CSP}^{d,k,t}_{n,\Delta}) = 2^{\Omega(n)}$. For $d^{k-1} > t \geq (d-1)(k-1)$, resolution complexity remains open. This generalizes the random k-SAT case in the following ways: for $k \geq 3$, to generalized satisfiability (i.e., domain size 2 but constraints with $t > 1$) and to domain sizes larger than two, and to binary CSPs. Table 1 summarizes what we know for the cases $k = 2$ and $d = 2$. Each cell shows a range of values of t for which given **C-RES** complexities hold – "?" denotes unknown. For $k = 2$, a selection of domain sizes is given, whereas for $d = 2$ a selection of constraint arities is given.

	$\mathbf{C\text{-}RES}(\mathsf{CSP}^{d,2,t}_{n,\Delta})$		
d	$2^{\Omega(n)}$	?	$O(1)$
3	1	2	3...9
4	1,2	3	4...16
5	1,2,3	4	5...25

(a) Binary CSPs ($k = 2$)

	$\mathbf{C\text{-}RES}(\mathsf{CSP}^{2,k,t}_{n,\Delta})$		
k	$2^{\Omega(n)}$	?	$O(1)$
3	1	2,3	4...8
4	1,2	3..7	8...16
5	1,2,3	4..15	16...32

(b) Generalized Satisfiability CSPs ($d = 2$)

Table 1. Resolution Complexity of Random CSPs: d=2 and $k = 2$ cases

The remainder of the paper is organized as follows. Section 2 gives basic definitions, and Section 3 develops the sub-critical expansion method for **C-RES** lower bounds. Section 4 applies this method to obtaining the claimed bounds for random CSPs, and Section 5 makes some concluding remarks.

2 Preliminaries

Since we have fixed domain size and constraint arity, we may adopt a slightly simpler definition of CSP instance than is usual.

Definition 1 (CSP terminology). *Let $d, k, t > 0$ be integers, and $\mathbf{R}$ be the set of all k-ary relations over the domain $[d] = \{1, \ldots, d\}$ having size exactly $(d^k) - t$ (i.e., tightness t). An instance $\mathcal{I}$ of the problem $\mathsf{CSP}^{d,k,t}$ is a set $\mathcal{I} = \{C_1, \ldots C_m\}$ of constraints. Each constraint C_i is an expression $R_i(\boldsymbol{x_i})$, where R_i is a symbol denoting some relation in $\mathbf{R}$, and $\boldsymbol{x_i}$ is a tuple of variable symbols. We denote by $vars(\mathcal{I})$ the set of variable symbols occurring in $\mathcal{I}$. An* assignment *for $CSP^{d,k,t}$ instance $\mathcal{I}$ is a function $\alpha : vars(\mathcal{I}) \mapsto [d]$. For variable tuple $\boldsymbol{x} = \langle x_1, \ldots, x_i \rangle$, we write $\alpha(\boldsymbol{x})$ for $\langle \alpha(x_1), \ldots, \alpha(x_i) \rangle$. Assignment α* satisfies *constraint $R(\boldsymbol{x}) \in \mathcal{I}$ if $\alpha(\boldsymbol{x})$ is in the relation denoted by R. Assignment α* satisfies *$\mathcal{I}$ if it satisfies* *every constraint of $\mathcal{I}$. $\mathcal{I}$ is satisfiable if there is an assignment that satisfies it. A* conflict *for instance $\mathcal{I}$ is a partial assignment α, defined on exactly the variables $\boldsymbol{x}$ for some constraint $R(\boldsymbol{x}) \in \mathcal{I}$ which does not satisfy that constraint. We write conflicts($\mathcal{I}$) to denote the set of all conflicts for $\mathcal{I}$.*

Definition 2 (propositional resolution). *A literal is a propositional variable or its negation, a clause is a set of literals, and a (CNF) formula is a set of clauses viewed as a conjunction of disjunctions under the usual semantics of propositional logic. The propositional* resolution rule *allows us to infer a clause $\{X\} \cup \{Y\}$ from two clauses $\{X\} \cup \{x\}$ and $\{Y\} \cup \{\overline{x}\}$ by resolving them on x. $\{X\} \cup \{Y\}$ is called the* resolvent. *A* resolution derivation *of a clause C from a set of clauses ϕ is a sequence $C_0, \ldots C_m$ of clauses, where $C_m = C$ and each C_i is either an member of ϕ or is derived by resolving two clauses C_j, C_k with $j, k < i$. The derivation is of length or size m. A resolution derivation of the empty clause (denoted $\Box$) from ϕ is called a* resolution refutation *of ϕ. We denote the propositional resolution system by **RES**.*

There is a resolution refutation of ϕ if and only if ϕ is unsatisfiable.

Definition 3 (CNF($\mathcal{I}$)). *To any $\mathsf{CSP}^{d,k,t}$ instance $\mathcal{I}$ we associate a formula CNF($\mathcal{I}$) as follows. For each CSP variable $v \in vars(\mathcal{I})$, we have d propositional variables, one for each value in $[d]$. We write $v\!:\!a$ for the propositional variable intended to assert that CSP variable $v \in vars(\mathcal{I})$ is assigned value a. Two sets of clauses encode properties of a satisfying assignment for $\mathcal{I}$: For each CSP variable $v \in vars(\mathcal{I})$, CNF($\mathcal{I}$) has a* domain clause *asserting that v must be assigned some value from $[d]$. For each constraint $R(\boldsymbol{y}) \in \mathcal{I}$, and each partial assignment α for exactly the variables in $\boldsymbol{y}$ and such that $\alpha(\boldsymbol{y}) \notin R$, CNF($\mathcal{I}$) has a* conflict clause *expressing that α is a conflict for $\mathcal{I}$.*

So, the propositional formula associated with $\mathcal{I}$ is

$$\mathrm{CNF}(\mathcal{I}) = \bigcup \begin{cases} \{\{v\!:\!a : a \in [d]\} : v \in \mathrm{vars}(\mathcal{I})\} \\ \{\{\overline{x\!:\!a} : x = a \in \alpha\} : \alpha \in \ \mathrm{conflicts}(\mathcal{I})\} \end{cases}$$

CNF($\mathcal{I}$) is satisfiable if and only if $\mathcal{I}$ is. Notice there is a one-to-one correspondence between conflict clauses in CNF($\mathcal{I}$) and elements of conflicts($\mathcal{I}$). A variant of this encoding includes clauses requiring each CSP variable to be assigned at most one value. All results in this paper are independent of these clauses.

Definition 4 (C-RES). *We define a* **C-RES** refutation *for any $\mathsf{CSP}^{d,k,t}$ instance $\mathcal{I}$ to be a propositional resolution (**RES**) refutation of CNF($\mathcal{I}$).*

Soundness and refutation completeness of **C-RES** follow from soundness and completeness of **RES**, and the fact that CNF($\mathcal{I}$) is satisfiable iff $\mathcal{I}$ is.

Definition 5 (resolution complexity). *The* resolution complexity *of a CNF formula ϕ, denoted **RES**(ϕ), is the minimum number of clauses in any resolution refutation of ϕ. The **C-RES** complexity a $\mathsf{CSP}^{d,k,t}$ instance $\mathcal{I}$, denoted **C-RES**($\mathcal{I}$), is the minimum number of clauses in any **C-RES** refutation of $\mathcal{I}$.*

Note that **C-RES**($\mathcal{I}$)=**RES**(CNF($\mathcal{I}$)). We extend **RES**() and **C-RES**() in the natural way to infinite sets. For example, to describe the **C-RES** complexity of a problem $\mathsf{CSP}^{d,k,t}$ we may write **C-RES**($\mathsf{CSP}^{d,k,t}$) $= O(f(n))$, denoting by $\mathsf{CSP}^{d,k,t}$ the set of all instances of problem $\mathsf{CSP}^{d,k,t}$ with n variables. Used this way **C-RES**($\mathsf{CSP}^{d,k,t}$) is implicitly a function of n.

3 C-RES Bounds *via* Sub-Critical Expansion

In this section we develop the notion of sub-critical expansion for CSPs, and its use in obtaining **C-RES** lower bounds.

Definition 6 (width, w(C), w(ϕ),w(π), W(ϕ)). *We define the* width $w(C)$ *of clause C, to be the number of literals in C, that is, the size of C. The width $w(\phi)$ of formula ϕ, is the size of the largest clause in ϕ. The width $w(\pi)$ of resolution refutation π is the width of the largest clause in π. Of special interest for a formula ϕ is the minimum width of any refutation of ϕ, $W(\phi) \stackrel{\text{def}}{=} \min\{w(\pi) : \pi \text{ is a refutation of } \phi\}$, which we call the* refutation width *of ϕ. We extend these functions to infinite families Φ: $w(\Phi) \stackrel{\text{def}}{=} \max\{w(\phi) : \phi \in \Phi_n\}$ and $W(\Phi) \stackrel{\text{def}}{=} \max\{W(\phi) : \phi \in \Phi_n\}$, where Φ_n is the set of n-variable members of Φ.*

The following result relates refutation width to the minimum refutation size. We give a slightly restricted form of the result sufficient for our context.

Theorem 1 (Ben-Sasson & Wigderson[6]). *For any set of formulas Φ with $w(\Phi)$ bounded by a constant, **RES**(Φ) $= 2^{\Omega((W(\Phi_n))^2/n)}$*

3.1 Width and Sub-Critical Expansion

Theorem 1 reduces lower bounds on **RES**(ϕ) to lower bounds on $W(\phi)$. By definition, lower bounds on **C-RES**($\mathcal{I}$) are shown by proving **RES** lower bounds for CNF($\mathcal{I}$). So we could obtain **C-RES**($\mathcal{I}$) bounds by deriving bounds on W(CNF($\mathcal{I}$)). This is less than ideal. For example it requires reasoning over arbitrary sub-formulas of CNF($\mathcal{I}$), which is not convenient because of the structure of CNF($\mathcal{I}$). So, in this section we reduce lower bounds on **C-RES**($\mathcal{I}$) to lower bounds on the sub-critical expansion of $\mathcal{I}$. In this way, we also make more transparent the properties of $\mathcal{I}$ that are relevant to intractability for **C-RES**.

To reason about minimum-size refutations, we need to start from minimum-sized unsatisfiable sub-instances.

Definition 7 (sub-instance). *An instance $\mathcal{J}$ is a* sub-instance *of $\mathcal{I}$ if and only if vars($\mathcal{J}$) $\subseteq$ vars($\mathcal{I}$) and the constraint set of $\mathcal{J}$ consists of all the constraints of $\mathcal{I}$ that involve only variables of $\mathcal{J}$.*

Definition 8 (critical size, $\mathcal{C}(\mathcal{I})$). *The* critical size *for instance $\mathcal{I}$, denoted $\mathcal{C}(\mathcal{I})$, is the largest integer such that every sub-instance $\mathcal{J}$ of $\mathcal{I}$ with $\mathcal{C}(\mathcal{I})$ or fewer variables is satisfiable.*

The following definition of a sub-instance implying a clause over the literals of CNF($\mathcal{I}$) makes a direct connection between sub-instances of $\mathcal{I}$ and clauses derived from CNF($\mathcal{I}$).

Definition 9 ($\widehat{\alpha}$, implies). *For any assignment α for $\mathcal{I}$, we denote by $\widehat{\alpha}$ the truth assignment for CNF($\mathcal{I}$) that assigns $x{:}a$ the value* **true** *iff $\alpha(x) = a$. We say that assignment α for $\mathcal{I}$ satisfies clause C over the literals of CNF($\mathcal{I}$) if $\widehat{\alpha}$ satisfies C. We say instance $\mathcal{J}$* implies *clause C if for every assignment α that satisfies $\mathcal{J}$, $\widehat{\alpha}$ satisfies C, in which case we write $\mathcal{J} \models C$.*

A "complex clause", for instance $\mathcal{I}$, is a clause that cannot be derived from CNF($\mathcal{J}$) for any very small sub-instance $\mathcal{J}$ of $\mathcal{I}$.

Definition 10 (clause complexity, $\mu(C)$). *For instance $\mathcal{I}$, and clause C over the literals of CNF($\mathcal{I}$), the* complexity of C *with respect to $\mathcal{I}$, denoted $\mu(C)$, is the size of the smallest sub-instance $\mathcal{J}$ of $\mathcal{I}$ such that $\mathcal{J} \models C$.*

The following lemma establishes that every **C-RES** refutation must contain a complex clause. (Curiously, its statement and proof match almost word-for-word identical those for the propositional case, although the terms actually mean something slightly different.)

Lemma 1. *For any $s \leq \mathcal{C}(\mathcal{I})$, every resolution refutation of CNF($\mathcal{I}$) contains a clause a clause satisfying $s \geq \mu(C) \geq s/2$.*

Proof. Let π be a **RES** refutation of CNF($\mathcal{I}$), and $s \leq \mathcal{C}(\mathcal{I})$. Consider the graph G_π of π. The vertices of G_π are the clauses in π, and there are two directed edges to each non-input clause A, one from each of the two clauses from which A is derived. Search G_π as follows. Let the "current" clause be $\Box$ (the root). Notice that $\mu(\Box) = \mathcal{C}(\mathcal{I})$. As long as at least one of clauses with an edge to the current clause has complexity greater than s, move to such a clause. Stop when both edges to the current clause are from clauses with complexity at most s. This must happen, since every input clause has complexity 1, and since the complexity of clauses on a path from root to leaf must be non-increasing. Now suppose that the current clause A is derived from clauses D and E. It is clear that $\mu(A) \leq \mu(D) + \mu(E)$, so not both of D and E can have complexity less that $s/2$, and one of them must be the desired clause. $\Box$

In discussing instance structure, it is useful to consider the following graph.

Definition 11 (constraint hypergraph/graph). *To every instance $\mathcal{I}$ we associate a hypergraph $G_{\mathcal{I}} = \langle V, E \rangle$ called the* constraint hypergraph *of $\mathcal{I}$, where $V = vars(\mathcal{I})$, and $E = \{vars(C) \mid C \in \mathcal{I}\}$.*

If $\mathcal{I}$ is binary, $G_{\mathcal{I}}$ is just a graph. Note that if $\mathcal{J}$ is a sub-instance of $\mathcal{I}$, then the constraint graph of $\mathcal{J}$ is an induced sub-graph of the constraint graph of $\mathcal{I}$.

Next we define the boundary of a sub-instance. Our notion of boundary is more general than that used in proofs for propositional resolution, as there boundary variables must occur only once, whereas our boundary variables may occur more than once, and even with both signs.

Definition 12 (neighborhood, $\mathcal{N}(\mathcal{I})$, boundary, $\mathcal{B}(\mathcal{I})$). *The* neighborhood *of a variable x in $\mathcal{I}$, denoted $\mathcal{N}(x)$, is the set of variables y such that there is a constraint $R(\boldsymbol{z})$ in $\mathcal{I}$, and both x and y occur in $\boldsymbol{z}$. The* boundary *of $\mathcal{I}$ is the set of variables x of $\mathcal{I}$ such that every assignment for $\mathcal{N}(x)$ that satisfies the sub-instance of $\mathcal{I}$ whose variables are $\mathcal{N}(x)$ can be extended to an assignment that satisfies the sub-instance with variables $\mathcal{N}(x) \cup \{x\}$. We denote the* size *of the boundary of $\mathcal{I}$ by $\mathcal{B}(\mathcal{I})$.*

We now draw a connection between boundary size and clause width.

Lemma 2. *Let C be a clause over the literals of CNF($\mathcal{I}$), and $\mathcal{J}$ a minimal sub-instance of $\mathcal{I}$ such that $\mathcal{J} \models C$. Then $w(C) \geq \mathcal{B}(\mathcal{J})$.*

Proof. Let x be any variable in $\mathcal{B}(\mathcal{J})$, and let $\mathcal{J}'$ be the largest sub-instance of $\mathcal{J}$ not including x. By minimality of $\mathcal{J}$ with respect to implying C, there is an assignment α for vars($\mathcal{J}$) such that α satisfies $\mathcal{J}'$ but $\widehat{\alpha}$ does not satisfy C. Denote by $\alpha[x\!=\!a]$ the assignment such that $\alpha[x\!=\!a](x) = a$, and $\alpha[x\!=\!a](y) = \alpha(y)$ if $y \neq x$. Clearly α satisfies the sub-instance of $\mathcal{J}$ whose variables are $\mathcal{N}(x)$, so there is a value a such that $\alpha[x\!=\!a]$ also satisfies $\mathcal{J}$. By choice of $\mathcal{J}$, $\widehat{\alpha[x\!=\!a]}$ satisfies C, so there must be a literal in C that is given different values by $\widehat{\alpha}$ and $\widehat{\alpha[x\!=\!a]}$. This must be a literal mentioning x, that is of the form $x\!:\!i$ or $\overline{x\!:\!i}$ for some value i, so C contains such a literal. Thus $w(C) \geq \mathcal{B}(()\mathcal{J})$. □

Finally, we need a measure for the boundary size of sub-critical sub-instances.

Definition 13 (sub-critical expansion, $\mathcal{E}(\mathcal{I})$). *The* sub-critical expansion *of $\mathcal{I}$, denoted $\mathcal{E}(\mathcal{I})$, is the largest integer such that, for some $s \leq \mathcal{C}(\mathcal{I})$, every sub-instance $\mathcal{J}$ with $s > |vars(\mathcal{J})| \geq s/2$, satisfies $\mathcal{B}(\mathcal{J}) \geq \mathcal{E}(\mathcal{I})$.*

Lemma 3. *For any instance $\mathcal{I}$, $W(\mathcal{I}) \geq \mathcal{E}(\mathcal{I})$.*

Proof. Let π be a **RES** refutation of CNF($\mathcal{I}$), and $s \leq \mathcal{C}(\mathcal{I})$. By Lemma 1 π has a clause C satisfying $s \geq \mu(C) \geq s/2$. By Lemma 2 and the definition of $\mu(C)$, $w(C) \geq \mathcal{B}(\mathcal{J})$ for some sub-instance $\mathcal{J}$ of $\mathcal{I}$ satisfying $s \geq |\text{vars}(\mathcal{J})| \geq s/2$. By definition of $\mathcal{E}(\mathcal{I})$, $\mathcal{B}(\mathcal{J}) \geq \mathcal{E}(\mathcal{I})$, so π contains a clause C with $w(C) \geq \mathcal{E}(\mathcal{I})$. □

Finally, the result we wanted in this section now follows immediately from Lemma 3 and Theorem 1.

Theorem 2. *For any set* $\mathbf{I}$ *of* $\mathsf{CSP}^{d,k,t}$ *instances,* ***C-RES***($\mathbf{I}$)$= 2^{\Omega(\mathcal{E}(\mathbf{I})^2/n)}$

4 Random CSP Results

We now apply Theorem 2 to obtain **C-RES** lower bounds for random CSPs.

Definition 14 ($\mathsf{CSP}^{d,k,t}_{n,\Delta}$). *Let d, k, t, n be positive integers, with $1 \leq t \leq d^k$, and let $\Delta > 0$ be a real. We denote by $\mathsf{CSP}^{d,k,t}_{n,\Delta}$ the random instance of $\mathsf{CSP}^{d,k,t}$ with $\lfloor \Delta n \rfloor$ constraints over n variables, selected uniformly at random from all such instances.*

$\mathcal{I} = \mathsf{CSP}^{d,k,t}_{n,\Delta}$ can be seen as being constructed by selecting $\lfloor \Delta n \rfloor$ constraints independently and uniformly at random from all possible constraints for a $\mathsf{CSP}^{d,k,t}$ instance.

Theorem 3. *Let $\Delta > 0$ be constant, and let $d \geq 2$, $k \geq 2$, and $k + d > 4$. Then for $1 \leq t < (d-1)(k-1)$, a random $\mathsf{CSP}^{d,k,t}_{n,\Delta}$ instance $\mathcal{I}$ almost surely satisfies* ***C-RES***$(\mathcal{I}) = 2^{\Omega(n)}$

The remainder of this section will comprise the proof of this. To obtain **C-RES** lower bounds using Theorem 2, we must show that we have large sub-critical expansion, which entails first showing that the critical size is large, and then that sub-instances not much smaller than the critical size have large boundaries. Both steps use a property of random hypergraphs.

Let k, n, m be positive integers. The random hypergraph denoted $H^k_{n,m}$ is the k-uniform hypergraph on n vertices generated as follows. The vertices of H are $V = [n]$. The edges E of H consist of m edges, selected independently and uniformly at random from the set of $\binom{n}{k}$ possible edges of size k. Notice that the constraint hypergraph of $\mathsf{CSP}^{d,k,t}_{n,\Delta}$ is the random hypergraph $H^k_{n,\Delta n}$.

Definition 15 **($c(\Delta, k, \epsilon)$).** *For any integer $k > 1$, and reals $\Delta, \epsilon > 0$, we define $c(\Delta, k, \epsilon) \stackrel{\text{def}}{=} \left(3(\mathbf{e}^k \Delta)^{\frac{1+\epsilon}{k-1}}\right)^{-1/\epsilon}$ where $\mathbf{e}$ denotes the base of the natural logarithm.*

Lemma 4. *Let $\Delta > 0$ be a real number, $k > 0$ an integer, and $H = H^k_{n,\Delta n}$ be the random k-uniform hypergraph with n vertices and $\lfloor \Delta n \rfloor$ edges. Then for any $\epsilon > 0$, a.s. no set of r vertices of H, for any $k \leq r \leq c(\Delta, k, \epsilon)n$, induces more than $(\frac{1+\epsilon}{k-1})r$ edges of H.*

This is a variation on Lemma 1 from [9]. We give a self-contained proof in Appendix A. Using this property, we can bound the critical size as follows.

Lemma 5. *Let $k, d \geq 2$, $1 \leq t < (k-1)(d-1)$, and $0 < \epsilon < 1/t$. Then a.s. $\mathcal{C}(\mathsf{CSP}^{d,k,t}_{n,\Delta}) \geq c(\Delta, k, \epsilon)n$*

A small lemma will be useful in proving this. Recall that a conflict for instance $\mathcal{I}$ corresponds to a conflict clause of CNF($\mathcal{I}$).

Definition 16 **(minimally unsatisfiable).** *A CSP instance $\mathcal{I}$ with n variables is minimally unsatisfiable if $\mathcal{I}$ is unsatisfiable but every sub-instance of $\mathcal{I}$ with fewer than n variables is satisfiable.*

Lemma 6. *Any minimally unsatisfiable instance $\mathcal{I}$ with s variables must have more than $(d-1)s$ conflicts.*

Proof. Suppose $\mathcal{I}$ is a minimally unsatisfiable CSP instance, with s variables and only $s(d-1)$ conflicts, and let ϕ denote CNF($\mathcal{I}$). ϕ is unsatisfiable if and only if $\mathcal{I}$ is. ϕ has exactly ds variables and $s(d-1) + s = ds$ clauses. Every minimally unsatisfiable set of clauses has more clauses than variables [3], so ϕ

is not minimally unsatisfiable. Let ψ be a minimally unsatisfiable subset of ϕ. Then ψ is a proper subset of ϕ, and since it has fewer clauses than ϕ, it must also have fewer variables than ϕ. This can only be the case if one of the clauses in $\phi - \psi$ is a domain clause. Suppose that such a domain clause is for the CSP variable x. Any conflict clause involving x contains a literal $\overline{x\!:\!i}$, and since the only occurrence of $x\!:\!i$ is in the domain clause, this literal would be a pure literal in ψ. Since pure literals cannot occur in a minimally unsatisfiable clause set, ψ must not contain any conflict clause involving x. Now, let $\mathcal{J}$ be the largest sub-instance of $\mathcal{I}$ that does not have variable x, and let ψ' denote $\mathrm{CNF}(\mathcal{J})$. Then $\psi \subseteq \psi'$, so $\mathcal{J}$ is unsatisfiable, contradicting the assumption that $\mathcal{I}$ is minimally unsatisfiable. □

Proof. (Lemma 5) By Lemma 6, any unsatisfiable sub-instance of $\mathcal{I}$ with s variables has a sub-instance with $r \le s$ variables and more than $(d-1)r$ conflicts. So a.s. no sub-instance of $\mathcal{I}$ with $s \le c(\Delta, k, \epsilon)n$ variables is unsatisfiable unless there is such a sub-instance with more than $(d-1)s$ conflicts, and thus with

$$\begin{aligned} t(\tfrac{1+\epsilon}{k-1})s > (d-1)s &\Rightarrow t(\tfrac{1+\epsilon}{k-1}) > (d-1) \\ &\Rightarrow t > (d-1)/(\tfrac{1+\epsilon}{k-1}) \\ &\Rightarrow t > (k-1)(d-1)/(1+\epsilon) \\ &\Rightarrow t \ge (k-1)(d-1), \text{ provided } \epsilon < 1/t. \end{aligned}$$

□

The dependence of Lemma 5 on the ratio of clauses to variables implies that this method of bounding critical size cannot work for higher values of t. As we will see below, when $k > 2$ an argument establishing large boundary size works for values of t up to (and indeed above) the value d^{k-1}, above which we have $O(1)$ upper bounds. Thus, extending our lower bounds to larger values of t requires a sharper method for lower bounds on critical size.

We next establish the boundary size for $\mathsf{CSP}^{d,k,t}_{n,\Delta}$. The following simple probabilistic lemma will be useful in doing this.

Lemma 7. *In an experiment in which n independent coins are tossed, each with fixed probability p, $0 < p < 1$, of producing a head, then for any constant $0 < c < 1$, the number of heads is a.s. at least cnp.*

Proof. A standard application of Chernoff bounds.

It will be convenient to treat the cases of $k = 2$ and $k \ge 3$ separately, and we begin with $k \ge 3$, which is slightly simpler.

Lemma 8. *Let $k \ge 3$, $1 \le t \le (d-1)d^{k-1}$, $0 < \epsilon < (2(k-1)/k) - 1$, and $0 < r < c(\Delta, k, \epsilon)$, all constant. Then for some $c > 0$, a.s. every sub-instance $\mathcal{J}$ of $\mathsf{CSP}^{d,k,t}_{n,\Delta}$ with rn variables has $\mathcal{B}(\mathcal{J}) \ge cn$.*

Proof. Let $r < c(\Delta, k, \epsilon)$, $s = rn$. Almost surely every size s sub-graph of $H = H^k_{n,\Delta n}$ has at most $s((1+\epsilon')/(k-1))$ edges, for any $\epsilon' \ge \epsilon$. So the total degree (sum of all vertex degrees) of such a sub-graph is at most $ks((1+\epsilon')/(k-1))$. The number of vertices of degree 2 or more must be at most 1/2 this value, so the number of vertices of degree 0 or 1 is at least

$$\begin{aligned} s - s(\tfrac{k(1+\epsilon')}{2(k-1)}) &= s(1 - \tfrac{k(1+\epsilon')}{2(k-1)}) \\ &= s(1 - \tfrac{k}{2(k-1)} - \epsilon' k/2(k-1)) \\ &= s(1 - \tfrac{k}{2(k-1)} - \epsilon), \text{ where } \epsilon = \epsilon' k/2(k-1) < \epsilon'. \end{aligned}$$

This is a positive constant provided that $\epsilon < (2(k-1)/k)-1$, so for some $c_1 > 0$, a.s. every size-r sub-instance has $c_1 n$ variables with at most one constraint.

Let x be a variable in $\mathcal{J}$ with degree 0 or 1 in $G_{\mathcal{J}}$, the constraint hypergraph for $\mathcal{J}$. Call x good if it is in the boundary of $\mathcal{J}$. If x has degree 0 it is good. If x has degree 1, there is a constant $p > 0$ such that x is good with probability at least p. To see this, let $R(\boldsymbol{z})$ be the constraint on x, and $\boldsymbol{y}$ be the $k-1$ variables in $\boldsymbol{z}$ other than x. We need only that at least one choice of t conflicts for R allows any assignment for $\boldsymbol{y}$ to be extended to x, and for this it is sufficient to have $t \leq (d-1)d^{k-1}$. This is so, since $(d-1)d^{k-1}$ conflicts can be chosen so that for each of the d^{k-1} assignments for $\boldsymbol{y}$ there are exactly $d-1$ conflicts agreeing with this assignment on $\boldsymbol{y}$, leaving one value of x free.

If x,y are two degree-1 variables in $\mathcal{J}$, the events that they are good are dependent only if x and y are in the same constraint. Consider the set of degree-1 variables one-by-one in some order, and for each delete from the set the other at most $k-1$ variables in the same constraint. We started with at least $c_1 n$ vertices, so are left with at least cn/k degree-1 vertices which are independent with respect to being good. By Lemma 7, for constant $1 > c_2 > 0$, a.s. $c_2 p c_1 n/k$ of these vertices are good. Let $c = c_2 p c_2/k$ and we are done. $\square$

This gives us boundaries of size cn, for constant $c > 0$, for sub-critical sized sub-instances with $\Theta(n)$ variables, for every value of $k \geq 3$. For $k = 2$ (binary problems) this proof gives us nothing. We need a slightly different approach, which takes advantage of the number of vertices of degree 2 or less.

Lemma 9. *Let $\epsilon > 0$. Then, in the random graph $H^2_{n,\Delta n}$, a.s. every set of at most $s < c(\Delta, 2, \epsilon)n$ vertices has at least $(1/3 - \epsilon)s$ vertices of degree 2 or less.*

Proof. By Lemma 4, a.s. every size $s \leq c(\Delta, 2, \epsilon)n$ sub-graph of $H = H^2_{n,\Delta,n}$ has at most $(1+\epsilon')s$ edges, for any $\epsilon' > \epsilon$. So the total degree of such graphs is at most $2(1+\epsilon')s$, and the number of vertices with degree 3 or more is at most $2(1+\epsilon')s/3$. Thus, the number of vertices with degree 2 or less is at least

$$\begin{aligned} s - \left(\tfrac{2s(1+\epsilon')}{3}\right) &= s(1 - \tfrac{2(1+\epsilon')}{3}) \\ &= s(1 - \tfrac{2}{3} - \tfrac{2\epsilon'}{3}) \\ &= s(1/3 - \epsilon), \text{ where } \epsilon = 2\epsilon'/3 < \epsilon'. \end{aligned}$$

$\square$

Not all of these $(1/3 - \epsilon)s$ vertices will be boundary vertices, unless we can guarantee that the constraints on them do not place conflicts on all values. For this, we would require that the total number of conflicts on a variable be strictly less than d, which would require $t < d/2$. However, we can do better by observing that for $t \geq d/2$, while some degree 2 vertices might have more than d conflicts, and therefore not be in the boundary, almost surely some large fraction are.

Lemma 10. *Let $d \geq 3$, $1 \leq t \leq d(d-1)$, $0 < \epsilon < 1/3$, and $0 < r \leq c(\Delta, k, \epsilon)$. Then for some $c > 0$, a.s. every sub-instance $\mathcal{J}$ of $\mathsf{CSP}^{d,2,t}_{n,\Delta}$, with size rn has $\mathcal{B}(\mathcal{J}) \geq cn$.*

Proof. There are at least $(1/3 - \epsilon)rn$ vertices with degree 2 or less. Call such a vertex good if it has degree less than 2 or has degree 2 and at least one value with no conflicts. Suppose x is a degree 2 vertex, with constraints $R_1(x, y)$ and $R_2(x, z)$. The probability that x is good is a positive constant provided there is at least one choice of the t conflicts for each of R_1 and R_2 that leave one value for x without conflicts. This is possible if $t \leq d(d-1)$, since we can pick a value for x to have no conflicts, with the other $d - 1$ values for x having d conflicts with y (one for each value) and d conflicts with z. Therefore, with some fixed probability $p > 0$, each of the $(1/3 - \epsilon)rn$ vertices with degree at most 2 is good. By Lemma 7, for every $0 < c_1 < 1$ a.s. at least $c_1(1/3 - \epsilon)rn$ of these variables are good, and these are in the boundary. Let $c = c_1(1/3 - \epsilon)r$. $\square$

Notice that this property does not hold for $t > d(d-1)$, since if x has $d(d-1) + 1$ conflicts with some other variable y, at least one value for y has conflicts with every value for x.

Lemma 11. *For any $d, k \geq 2$ with $d + k > 4$, any $\Delta > 0$, and any t, $1 \leq t \leq (d-1)d^{k-1}$, there is a constant $c > 0$ such that a.s. $\mathcal{E}(\mathsf{CSP}^{d,k,t}_{n,\Delta}) \geq cn$.*

Proof. Let $r < c(\Delta, k, \epsilon)$, so that a.s. all instances of size at most rn are satisfiable. Then let $\epsilon > 0$ be small enough to satisfy the conditions of Lemma 5 and Lemma 8, and $c > 0$ satisfy the conditions of either Lemma 8 or Lemma 10, depending on k, and we are done. $\square$

The lower bounds now follow immediately from Lemma 11 and Theorem 2.

Proof. (Theorem 3) Let $d \geq 2$, $k \geq 2$, be integer constants with $d + k > 4$, and let t satisfy $(d-1)(k-1) > t > 0$. By Lemma 11, for some constant $c > 0$, a.s., $\mathcal{E}(\mathsf{CSP}^{d,k,t}_{n,\Delta}) \geq cn$. By Theorem 2, a.s. **C-RES**(**I**)$= 2^{\Omega(\mathcal{E}(\mathbf{I})^2/n)}$. From this it follows that almost surely **C-RES**$(\mathsf{CSP}^{d,k,t}_{n,\Delta,}) = 2^{\Omega(c^2 n)}$ $\square$

5 Conclusions

We have presented two important properties for the complexity of CSP refutations in a (useful if not entirely natural) resolution-like system for CSPs. The first is that a (not completely non-intuitive) structural property of CSP instances, the sub-critical expansion, is closely related to the resolution complexity. The second is that for an interesting range of parameters, typical random CSP instances almost surely have linear size sub-critical expansion, and thus almost surely require exponential size refutations.

The refutation system used is interesting mainly because it can efficiently simulate the reasoning implicit in almost all current complete CSP algorithms. Instances which are hard for this refutation system must also be hard for these algorithms. It follows that in certain parameter ranges, almost all CSP instances require exponential time for these algorithms (since a uniformly selected one does!).

We used the sub-critical expansion property in obtaining our random CSP results. However, it should also be applicable to many non-random families of CSPs. Moreover, it provides considerable insight into the structural properties of instances that lead to intractability for current methods. Even for families of instances for which one would not or could not carry out a lower bound proof, it may provide some intuition about when these are likely to be hard for resolution-based algorithms.

References

1. D. Achlioptas, P. Beame, and M. Molloy. A sharp threshold in proof complexity. In *Proc., 33rd Annual ACM Symp. on the Theory of Computing (STOC-01)*, pages 337–346, 2001.
2. D. Achlioptas, L. Kirousis, E. Kranakis, D. Krizanc, M. Molloy, and Y. Stamatiou. Random constraint satisfaction: A more accurate picture. *Constraints*, 6(4):329–344, 2001.
3. R. Aharoni and N. Linial. Minimal unsatisfiable formulas and minimal non-two-colorable hypergraphs. *J. of Combinatorial Theory, Series A*, 43:196–204, 1986.
4. Andrew B. Baker. *Intelligent Backtracking on Constraint Satisfaction Problems: Experimental and Theoretical Results*. PhD thesis, University of Oregon, 1995.
5. P. Beame, R. Karp, T. Pitassi, and M. Saks. On the complexity of unsatisfiability proofs for random k-CNF formulas. In *Proc., 30th Annual ACM Symp. on the Theory of Computing (STOC-98)*, pages 561–571, May 1998.
6. E. Ben-Sasson and A. Wigderson. Short proofs are narrow: Resolution made simple. In *Proc., 31st Annual Symp. on the Theory of Computing (STOC-99)*, pages 517–526, May 1999. (Also appears as ECCC report TR99-022).
7. P. Cheesman, B. Kanefsky, and W. M. Taylor. Where the really hard problems are. In *Proc., 12th Int'l. Joint Conf. on A. I. (IJCAI-91)*, 1991.
8. V. Chvátal and B. Reed. Mick gets some (the odds are on his side). In *Proc. of the 33rd Symp. on the Foundations of Comp. Sci. (FOCS-92)*, pages 620–628, 1992.
9. V. Chvátal and E. Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, 1988.
10. J. De Kleer. A comparison of ATMS and CSP techniques. In *Proc. of the 11th Int'l. Joint Conf. on A. I. (IJCAI-89)*, pages 290–296, 1989.
11. R. Dechter. Enhancement schemes for constraint processing: Backjumping, learning, and cutset decomposition. *Artificial Intelligence*, 41:273–312, 1990.
12. R. Dechter and D. Frost. Backtracking algorithms for constraint satisfaction problems. Technical report, Dept. of Inf. & Comp. Sci., U. of California, Irvine, 1999.

13. J. Gashnig. Experimental case studies of backtrack vs. Waltz-type vs. new algorithms for satisficing assignment problems. In *Proc. of the Canadian Artificial Intelligence Conference*, pages 268–277, 1978.
14. I.P. Gent, E. MacIntyre, P. Prosser, B.M. Smith, and T. Walsh. Random constraint satisfaction: flaws and structure. *Constraints*, 6(4):345–372, October 2001.
15. O. Kullmann. Upper and lower bounds on the complexity of generalized resolution and generalized constraint satisfaction problems. Submitted, 2000.
16. A. K. Mackworth and E. C. Freuder. The complexity of some polynomial network consistency algorithms for constraint satisfaction problems. *Artificial Intelligence*, 25:65–73, 1985.
17. D. G. Mitchell. Hard problems for CSP algorithms. In *Proc., 15th Nat. Conf. on Artificial Intelligence (AAAI-98)*, pages 398–405, 1998.
18. D. G. Mitchell. *The Resolution Complexity of Constraint Satisfaction.* PhD thesis, University of Toronto, 2002.
19. D. G. Mitchell. Constraint satisfaction and resolution. In preparation.
20. D. G. Mitchell, B. Selman, and H. J. Levesque. Hard and easy distributions of SAT problems. In *Proc. of the 10th Nat. Conf. on A. I. (AAAI'92)*, pages 459–462, 1992.
21. M. Molloy. Models and thresholds for random constraint satisfaction problems. In *Proc., 34th Annual Symp. on the Theory of Computing (STOC-02)*, pages 209–217. ACM, 2002.
22. P. Prosser. Hybrid algorithms for the constraint satisfaction problem. *Computational Intelligence*, 9(3):268–299, August 1993.
23. P. Prosser. An empirical study of phase transitions in binary constraint satisfaction problems. *Artificial Intelligence*, 81:81–109, 1996.
24. T.J. Schaefer. The complexity of satisfiability problems. In *Proc., 10th Annual ACM Symp. on the Theory of Computing (STOC-78)*, pages 216–226. ACM, 1978.

A Local Sparsity of Random Hypergraphs

Here we give a proof of Lemma 4: If $c(\Delta, k, \epsilon) = (3(\mathbf{e}^k \Delta)^{\frac{1+\epsilon}{k-1}}))^{-1/\epsilon}$, and $\epsilon > 0$ then a.s. $H^k_{n,m}$ has no set of r vertices, for any $k \le r \le c(\Delta, k, \epsilon)n$, that induce more than $r(1+\epsilon)/(k-1)$ edges. For the remainder, H will always denote the random hypergraph $H^k_{n,m}$. Call H *(r,s)-dense* if some set of r vertices of H induces s or more edges of H.

Lemma 12. *If $r, s \ge 1$, then $\mathbf{Pr}[H$ is (r,s)-dense$] \le \left(\frac{\mathbf{e}n}{r}\right)^r \left(\frac{\mathbf{e}mr^k}{sn^k}\right)^s$*

Proof. Let $H = \langle V, E\rangle$, and $R \subseteq V$ be a set of vertices of H of size r. The probability that a randomly chosen edge on V is also an edge on R is $\binom{r}{k}/\binom{n}{k}$. The probability that at least s edges of H are induced by R is the probability that some size-s subset of E is on R, which is at most the expected number of size-s subsets of E that have all of their vertices in R, so we have $\mathbf{Pr}[H$ has s edges in $R] \le \binom{m}{s}\left(\binom{r}{k}/\binom{n}{k}\right)^s \le \left(\frac{\mathbf{e}m}{s}\right)^s \left(\frac{r}{n}\right)^{ks} \le \left(\frac{\mathbf{e}mr^k}{sn^k}\right)^s$

Summing over all $\binom{n}{r}$ possible choices for R, we obtain

$$\mathbf{Pr}[H \text{ is } (r,s) - \text{dense}] \le \binom{n}{r}\left(\frac{\mathbf{e}mr^k}{sn^k}\right)^s \le \left(\frac{\mathbf{e}n}{r}\right)^r \left(\frac{\mathbf{e}mr^k}{sn^k}\right)^s$$

$\square$

Lemma 13. *Let $z > 0$ be a real, and $s \le n \in \mathbb{N}$. The probability that, for some r, $1 \le r \le s$, $H^k_{n,m}$ is (r, zr)-dense is at most* $\sum_{r=1}^{s} \left(\mathbf{e}n \left(\frac{\mathbf{e}m}{zn^k}\right)^z r^{z(k-1)-1}\right)^r$

Proof. $\mathbf{Pr}[\exists r \le s \text{ s.t. } H \text{ is } (r, zr)\text{-dense}] \le \sum_{r=1}^{s}$, and

$$\sum_{r=1}^{s} \left(\frac{\mathbf{e}n}{r}\right)^r \left(\frac{\mathbf{e}mr^k}{zrn^k}\right)^{zr} = \sum_{r=1}^{s} \left(\frac{\mathbf{e}n}{r}\left(\frac{\mathbf{e}mr^k}{zrn^k}\right)^z\right)^r = \sum_{r=1}^{s} \left(\mathbf{e}n\left(\frac{\mathbf{e}m}{zn^k}\right)^z r^{z(k-1)-1}\right)^r$$

□

Proof. (Lemma 4) Say that H is bad if for some choice of r, $1 \le r \le c(\Delta, k, \epsilon)n$, H is $(r, r(1+\epsilon)/(k-1))$-dense. We need to show H is a.s. not bad. Let z denote $(1+\epsilon)/(k-1)$ and s denote $c(\Delta, k, \epsilon)n$. Then we have

$$\mathbf{Pr}[H \text{ is bad }] \le \sum_{r=1}^{s} \left(\mathbf{e}n \left(\frac{\mathbf{e}\Delta n}{zn^k}\right)^z r^{z(k-1)-1}\right)^r$$

Now we write A for $\mathbf{e}n(\frac{\mathbf{e}\Delta n}{zn^k})^z$ and write $B(r)$ for $(Ar^{z(k-1)-1})^r$, and have that $\mathbf{Pr}[H \text{ is bad }] \le \sum_{r=1}^{s} B(r)$, which is a geometric series in r. If $B(r+1)/B(r) \le 1/2$, the series will be less than twice its largest term, so we have that

$$\begin{aligned}
\mathbf{Pr}[H \text{ is bad }] &\le 2B(1)\\
&= 2\mathbf{e}n\left(\tfrac{\mathbf{e}\Delta n}{zn^k}\right)^z\\
&= 2(1/z)^z\mathbf{e}(\mathbf{e}\Delta)^z n^{z(1-k)+1}\\
&\le 3\mathbf{e}(\mathbf{e}\Delta)^{(1+\epsilon)/(k-1)} n^{(1-k)(1+\epsilon)/(k-1)+1}\\
&= 3\mathbf{e}(\mathbf{e}\Delta)^{(1+\epsilon)/(k-1)} n^{-\epsilon}\\
&= o(1)
\end{aligned}$$

It remains to verify the condition on B:

$$\begin{aligned}
\tfrac{B(r+1)}{B(r)} &= \tfrac{(A(r+1)^{z(k-1)-1})^{r+1}}{(Ar^{z(k-1)-1})^r}\\
&= A\left(\left(\tfrac{r+1}{r}\right)^r (r+1)\right)^{z(k-1)-1}\\
&\le A\left(\mathbf{e}(r+1)\right)^{z(k-1)-1} && (\text{since } (\tfrac{x+1}{x})^x \le \mathbf{e})\\
&= \mathbf{e}n\left(\tfrac{\mathbf{e}\Delta n}{zn^k}\right)^z \left(\mathbf{e}(r+1)\right)^{z(k-1)-1}\\
&= \left(\tfrac{1}{z}\right)^z \left(\mathbf{e}^k\Delta\right)^z n^{z(1-k)+1}(r+1)^{z(k-1)-1}\\
&\le 1.5\left(\mathbf{e}^k\Delta\right)^z \left(\tfrac{r+1}{n}\right)^{z(k-1)-1} && (\text{since } \forall z > 0, (1/z)^z < 1.5)\\
&\le 1.5\left(\mathbf{e}^k\Delta\right)^z \left(\tfrac{c(\Delta,k,\epsilon)n}{n}\right)^{z(k-1)-1}\\
&= 1.5\left(\mathbf{e}^k\Delta\right)^z c(\Delta,k,\epsilon)^{z(k-1)-1}\\
&= 1.5\left(\mathbf{e}^k\Delta\right)^z \left(\left(3(\mathbf{e}^k\Delta)^{\frac{1+\epsilon}{k-1}}\right)^{-1/\epsilon}\right)^{z(k-1)-1}\\
&= 1.5\left(\mathbf{e}^k\Delta\right)^{\frac{1+\epsilon}{k-1}} \left(3(\mathbf{e}^k\Delta)^{\frac{1+\epsilon}{k-1}}\right)^{-1}\\
&= 1/2
\end{aligned}$$

□