A Random Constraint Satisfaction Problem That Seems Hard for DPLL

Harold Connamacher

University of Toronto
Department of Computer Science
10 King's College Road
Toronto, Ontario M5S 3G4
Canada
hsc@cs.toronto.edu

Abstract. This paper discusses an NP-complete constraint satisfaction problem which appears to share many of the threshold characteristics of SAT but is similar to XOR-SAT and so is easier to analyze. For example, the exact satisfiability threshold for this problem is known, and the problem has high resolution complexity. In this paper, we prove the problem appears hard for DPLL. Specifically, if we pick a problem instance at random with constraint density higher than some given threshold but below the satisfiability threshold, a DPLL backtracking algorithm using the unit clause heuristic will, with uniformly positive probability, take exponential time to find a satisfying assignment.

1 Introduction

The satisfiability threshold conjecture is that there exists a value c^* such that a random SAT formula on n variables and cn clauses, with n tending to infinity, is almost surely unsatisfiable if $c > c^*$ and almost surely satisfiable if $c < c^*$. From experimental evidence, the threshold for 3-SAT is $c^* \approx 4.2$ [1, 2]. Computer scientists have yet to prove this, and the current state of the research has 3.52 [3, 4] $\leq c^* \leq 4.506$ [5]. Even the existence of any exact threshold c^* has not been proven but something close has [6].

The satisfiability threshold is known for 2-SAT [7–9], and considering formulae with a mixture of 2- and 3-clauses (the $(2+p)$-SAT model introduced by [10]), an exact threshold is known if at most $\frac{2}{5}$ of the clauses are 3-clauses. For more than $\frac{2}{5}$ 3-clauses, a range for the threshold, if it exists, is given [11].

Researchers noticed that problems drawn from near the satisfiability threshold appear to require exponential time to satisfy while problems drawn from well below the threshold can be solved quickly [1]. It is known that if $c > 3.81$ DPLL with the unit clause heuristic will take exponential time [12] while for $c < 2.66$ the running time of the algorithm will be linear [13]. For generalized unit clause, the range is 3.003 [14] $\leq c \leq 3.98$ [12]. Experimental evidence supports the lower bounds as the threshold for linear versus exponential running time of DPLL with either the unit clause or generalized unit clause heuristic [15].

Many other NP-complete problems have similar threshold phenomena, and in statistical mechanics, these problems correspond to spin-glass like models over random graphs at zero temperature [16]. The simplest non-trivial such model is called the p-spin model (or k-spin in the notation of this paper) and at the zero temperature, this is k-XOR-SAT [17]. In XOR-SAT, a solution must satisfy an exclusive-or on each clause, rather than a disjunction, and being a simpler model than SAT, XOR-SAT appears to be easier to analyze. For example, the satisfiability threshold for 3-XOR-SAT is known [17, 18]. In this paper, we present an NP-complete constraint satisfaction problem which generalizes XOR-SAT. As a result, all the theorems proven in this paper also apply to XOR-SAT. For example, even though XOR-SAT is in P, a corollary of the main theorem of this paper is that at worst case DPLL with the unit clause heuristic will need exponential time to find a solution.

If we consider XOR-SAT as a constraint satisfaction problem with constraints on k variables, the key property which makes XOR-SAT simple to analyze is that each constraint is uniquely extendible. That is, for each possible assignment to $k-1$ variables of a constraint, there is a

unique legal value for the kth variable. The NP-complete problem considered is the generic uniquely extendible constraint satisfaction problem (k, d)-UE-CSP. As with XOR-SAT, (k, d)-UE-CSP seems easier to analyze than SAT, and [19] gives the satisfiability threshold of $(3, d)$-UE-CSP, but unlike XOR-SAT, no known polynomial time algorithm exists which almost surely solves a satisfiable instance of (k, d)-UE-CSP.

In this paper, we prove that a DPLL backtracking algorithm using the unit clause heuristic will, with uniformly positive probability, take exponential time to find a satisfying assignment on a random instance of $(3, d)$-UE-CSP taken from below the satisfiability threshold. As part of the proof, we study threshold behavior for a UE-CSP with a mixture of clauses of size 2 and 3 similar to the $(2 + p)$-SAT model, and we prove theorems similar to the known results for $(2 + p)$-SAT. As a result, for $(3, d)$-UE-CSP, we have three threshold values similar to those known and conjectured for 3-SAT: $c^* > c_e \geq c_l$ where c^* is the satisfiability threshold, c_e is the lowest density for which we can prove DPLL with unit clause requires exponential time, and c_l is the highest density for which unit clause alone will find a satisfying assignment in linear time. For 3-SAT, it is conjectured that $c_e = c_l$, and thus it would be interesting if we could prove this equality holds for $(3, d)$-UE-CSP.

Although the techniques used in this paper are similar to those for proving comparable theorems about SAT, the results are not quite the same. For example, where it is possible to prove behavior almost surely with SAT, we often can only prove with uniformly positive probability for (k, d)-UE-CSP.

1.1 (k, d)-UE-CSP

In (k, d)-UE-CSP, each constraint is over a k-tuple of variables, each variable must take a value from the domain $\{0, \ldots, d-1\}$, and every constraint is uniquely extendible. Note that k-XOR-SAT is exactly $(k, 2)$-UE-CSP. We will denote the problem UE-CSP when we allow a mixture of clause sizes and when d may be an arbitrary constant. We always assume $d \geq 2$.

In this paper, each constraint of the CSP is called a *clause*, and constraints of size i are denoted *i-clauses*. In keeping with terminology for SAT, each instance of the CSP is called a *problem* or a *formula*.

For $d \leq 3$, the problem is in P, but in [19], the following theorems are proven.

Theorem 1. *$(3, 4)$-UE-CSP is NP-complete.*

Theorem 2. *The satisfiability threshold for random $(3, 4)$-UE-SAT is $c^* = .917935...$.*

Theorem 3. *For any constant $c > 0$, the resolution complexity of a uniformly random instance of $(3, 4)$-UE-SAT with n variables and cn clauses is almost surely $2^{\Theta(n)}$.*

Note that the threshold for $(3, 4)$-UE-CSP is exactly the same as for 3-XOR-SAT [17, 18]. In addition, this threshold is true for all $d \geq 3$.

2 Main Theorem

We write a sequence of events $\mathcal{E}_n$ holds *almost surely* (a.s.) if $\lim_{n\to\infty} \mathbf{Pr}(\mathcal{E}_n) = 1$ and $\mathcal{E}_n$ holds *with uniformly positive probability* (w.u.p.p.) if $\liminf_{n\to\infty} \mathbf{Pr}(\mathcal{E}_n) > 0$.

The Davis-Putnam-Logemann-Loveland (DPLL) algorithm forms the basis of most current complete SAT solvers. The algorithm is a simple backtracking framework. At each step, an unassigned variable v is assigned a value. Any clause which is satisfied by the assignment is removed, v is removed from any clauses in which it occurs, and the constraint on those clauses is appropriately modified. DPLL then recurses on this reduced formula. If a conflict occurs, DPLL backtracks and tries a different value for v. Note that for UE-CSP, only singleton clauses will be removed.

One of the variations in DPLL is in the procedure for choosing the next variable. A method which is used in many SAT solvers is the unit clause (UC) heuristic which states that if clauses of size 1 exist, the next chosen variable is from one of the singleton clauses. For simplicity of analysis, if no clause of size 1 exists, the next variable is chosen randomly.

Note 4. A key observation, see e.g. [20], is that until the algorithm backtracks, the subproblem produced at each step by UC is uniformly random. Specifically, the 2-clauses form a uniformly random instance of $(2, d)$-UE-CSP, and the 3-clauses form a uniformly random instance of $(3, d)$-UE-CSP.

In order to model the subformulae produced by a running of DPLL, we introduce the random $((2 + p), d)$-UE-CSP model, similar to the $(2 + p)$-SAT model. In this model, a UE-CSP instance on n variables and m constraints has pm clauses of size 3 and $(1 - p)m$ clauses of size 2.

Experiments suggest that there is an exact threshold for $(2+p)$-SAT, and if the search algorithm produces a subformula which falls on the unsatisfied side of the threshold, the algorithm will take a long time to backtrack out of the subformula [15]. This paper will prove (k, d)-UE-CSP has similar behavior.

The main theorem is as follows.

Theorem 5. *A DPLL algorithm using the unit clause heuristic will take exponential time w.u.p.p. on a uniformly random instance of* $(3, 4)$*-UE-CSP with* n *variables and at least* $\frac{8}{9}$ *clauses.*

Proof. From Lemma 11, w.u.p.p. the unit clause heuristic will guide DPLL to a subformula with n' variables, $\left(\frac{1}{2} - \epsilon\right) n'$ 2-clauses and $\beta n'$ 3-clauses where $\beta > \left(\frac{1}{2} + \epsilon\right)$, $n' \geq \delta n$ for some $\delta > 0$ and which is a uniformly random UE-CSP on those parameters. From Lemma 7, such a configuration is a.s. unsatisfiable. From Lemma 12, DPLL will require $2^{\Omega(n')}$ steps to backtrack out of this configuration, w.u.p.p. □

Theorem 5 is a stronger result than the corresponding theorems for SAT because in the SAT theorems, the threshold for exponential behavior is below the conjectured satisfiability threshold but above the proven lower bound for the satisfiability threshold.

3 Threshold Behavior for $((2 + p), d)$-UE-CSP

One technique for analyzing the structure of an instance C of a constraint satisfaction problem is to consider the underlying hypergraph H of C. We define H to have as vertices the variables of C, and a set of variables are joined in a hyperedge iff that set forms a constraint in C. As with clauses, we denote a hyperedge of size i as an *i-edge*.

One example of this technique is to show the satisfiability threshold for $(2, d)$-UE-CSP is $\frac{1}{2}$.

Lemma 6. *For* $c < \frac{1}{2}$*, a uniformly random instance* C *of* $(2, d)$*-UE-CSP with* n *variables and* cn *constraints is w.u.p.p. satisfiable, and for* $c > \frac{1}{2}$*, it is a.s. unsatisfiable.*

Proof. Consider the random graph H which is the underlying hypergraph of C. The proof follows from the well known property of random graphs on n vertices and cn edges. If $c < \frac{1}{2}$, H has a.s. at most a constant number of cycles, and if $c > \frac{1}{2}$, the number of cycles in H grows unbounded, a.s. From the observation that each cycle in H creates a possible conflict in C, the lemma follows. □

Let c_p be the satisfiability threshold for $((2 + p), d)$-UE-CSP, if it exists. To get a nontrivial upper bound for c_p, we count the expected number of solutions to a random instance of $((2+p), d)$-UE-CSP. For both $(2, d)$-UE-CSP and $(3, d)$-UE-CSP, a random assignment satisfies each clause with probability $\frac{1}{d}$. Thus, if $\mathcal{S}_n$ is the set of solutions for any $((2 + p), d)$-UE-CSP formula on n variables, the expected number of solutions is

$$\mathbf{Exp}(|\mathcal{S}_n|) = d^n \left(\frac{1}{d}\right)^{cn}$$

which is goes to 0 as n goes to infinity if $c > 1$. Thus, from Markov's Inequality, we get $c_p \leq 1$ and the following lemma.

Lemma 7. *For any* $\epsilon > 0$*, a uniformly random UE-CSP instance with* $\left(\frac{1}{2} - \epsilon\right) n$ *2-clauses and* βn *3-clauses with* $\beta > \left(\frac{1}{2} + \epsilon\right)$ *is a.s. unsatisfiable.*

A second technique we will use in the paper is to model the behavior of UC, without backtracking, by a system of differential equations. Let $C_i(t)$ be the number of i-clauses at step t of the algorithm. Note that at each step of the algorithm, an unassigned variable is given a value. Thus, if no backtracking occurs, the number of steps is the same as the number of assigned variables. Let x be the number of variables assigned a value, $c_i(x)$ the number of i-clauses with c_i and x normalized to the range $[0,1]$. Then using the same justifications as [11, 20] for the behavior of UC on 3-SAT, we have

$$\frac{dc_3}{dx} = -\frac{3c_3(x)}{(1-x)}$$
$$\frac{dc_2}{dx} = \frac{3c_3(x)}{(1-x)} - \frac{2c_2(x)}{(1-x)},$$

and, by a theorem of [21], for any $\epsilon > 0$ and for $0 \le t \le (1-\epsilon)n$, a.s.

$$C_i(t) = c_i(t/n) \cdot n + \mathrm{o}(n).$$

Solving the above differential equations gives

$$C_3(t) = c_3(0)(1-t/n)^3 \cdot n + \mathrm{o}(n) \quad (1)$$
$$C_2(t) = (c_2(0) + 3c_3(0)(t/n))(1-t/n)^2 \cdot n + \mathrm{o}(n) \quad (2)$$

The important observation is that as long as no clause of length 0 is generated, no contradiction is reached. A clause of length 0 will only be generated if we have more than one clause of length 1, and the expected number of clauses of length 1 generated at step t is $2C_2(t)/(n-t)$. So if this density is bounded by $(1-\delta)$ for some $\delta > 0$, we will not expect to generate contradictions. This observation is summarized in the following lemma which is a corollary of lemmas in [22] and [11, 20].

Lemma 8 ([11]). *Fix $\delta, \epsilon > 0$ and let $t_0 = n - \lfloor \epsilon n \rfloor$. If for all $0 \le t \le t_0$ a.s. $C_2(t) < \frac{1}{2}(1-\delta)(n-t)$ then w.u.p.p. $C_1(t_0) + C_0(t_0) = 0$.*

Thus we can use the differential equations (1) and (2) to a.s. trace the first $t_0 = n - \lfloor \epsilon n \rfloor$ steps of UC and bound the probability that UC fails. For Lemmas 9 and 10, we need to deal with the final $n - t_0$ steps. By Lemma 8, after step t_0 we are left with a formula with ϵn variables, w.u.p.p. no clauses of length 1, and a.s. $C_3(t_0) + C_2(t_0)$ clauses of length at least 2 where

$$\begin{aligned} C_3(t_0) + C_2(t_0) &= c_3(0)\epsilon^3 n + (c_2(0) + 3c_3(0)(1-\epsilon))\epsilon^2 n \\ &\le r\epsilon^2 n \end{aligned}$$

for some constant $r > 0$. Observe that we can pick ϵ small enough so that, by a similar random graph argument as Lemma 6, these remaining clauses will a.s. induce at most a constant number of cycles. Thus, w.u.p.p. UC will find a satisfying assignment.

Lemma 9. *For $p \le \frac{1}{4}$, $c_p = \frac{1}{2(1-p)}$.*

Proof. Plugging $c_3(0) = cp$ and $c_2(0) = c(1-p)$ into (1) and (2) and adding the bound that $C_2(t) < \frac{1}{2}(1-\delta)(n-t)$ gives

$$2c(3px - p + 1)(1-x) < 1 \quad (3)$$

where $x = \frac{t}{n}$.

Note that if $p \le \frac{1}{4}$, the l.h.s. of (3) is a decreasing function of x and thus the inequality holds iff it holds for $x = 0$, and plugging in $x = 0$ gives

$$c < \frac{1}{2(1-p)}.$$

Applying Lemma 8 and the above observation completes the proof. □

Likewise, by plugging $c_3(0) = c$ and $c_2(0) = 0$ into (1) and (2) gives the following lemma.

Lemma 10. *Let C be a uniformly random instance of $(3,d)$-UE-CSP with n variables and at most $\frac{2}{3}n$ clauses. Then w.u.p.p. DPLL with UC will find a satisfying assignment without backtracking.*

Finally, we use this technique to prove the following lemma.

Lemma 11. *Let C be a uniformly random instance of $(3,d)$-UE-CSP with n variables and $\frac{8}{9}$ clauses. Then w.u.p.p. DPLL with UC will reach a subproblem C' of C which has $n' > \frac{3}{4}n$ variables, $\left(\frac{1}{2}-\epsilon\right)n'$ 2-clauses and $\beta n'$ 3-clauses, $\beta > \left(\frac{1}{2}+\epsilon\right)$, with all such subproblems equally likely.*

Proof. Let $c_2(0) = 0$ and $c_3(0) = \Delta$ and find a $t' = \pi n$ such that $C_2(t') = \left(\frac{1}{2}-\epsilon\right)n$ and $C_3(t') > \left(\frac{1}{2}-\epsilon\right)n$.

Note that $C_3(t)$ is a decreasing function while $C_2(t)$ is initially an increasing function. So we find the point t_e at which $C_3(t_e) = C_2(t_e)$. This gives $1 - \frac{t_e}{n} = 3\frac{t_e}{n}$ and so $t_e = \frac{1}{4}n$. Plugging t_e into $C_3(t)$ yields $\frac{1}{2}\left(n - \frac{1}{4}n\right) = \Delta\left(1-\frac{1}{4}\right)^3 n$ and thus $\Delta = \frac{8}{9}$.

By observing that for $0 < x \leq \frac{t_e}{n}$, $\frac{dc_2}{dx} < -\frac{dc_3}{dx}$, we can pick $t' < t_e$ and get the desired result.

Note that $C_2(t) < \frac{1}{2}(n-t)$ for all $0 \leq t \leq t_e$. Thus, by Lemma 8 w.u.p.p. no conflict occurs implying DPLL will not backtrack before reaching this configuration, and thus by Note 4 this configuration is uniformly random over all such mixed formulae with these clause densities. □

4 Resolution Lower Bound

The final step to prove Theorem 5 is the following lemma.

Lemma 12. *For any $\Delta, \epsilon > 0$, DPLL will require $2^{\Omega(n)}$ steps w.u.p.p. to backtrack out of a uniformly random UE-CSP instance with $\left(\frac{1}{2}-\epsilon\right)n$ 2-clauses and Δn 3-clauses, if that instance is unsatisfiable.*

From techniques developed in [23–25], exponential running time for DPLL on an unsatisfiable formula is a consequence of the formula requiring an exponential size resolution proof of unsatisfiability, and the shortest resolution proof for a CSP has exponential size, a.s., if there exists constants $\alpha, \zeta > 0$ such that a.s. the following three conditions hold.

1. Every subproblem on at most αn variables is satisfiable.
2. Every subproblem on v variables where $\frac{1}{2}\alpha n \leq v \leq \alpha n$ has at least ζn variables of degree 1.
3. If x is a variable of degree 1 in a CSP f then, letting f' be the subproblem obtained by removing x and its constraint, any satisfying assignment of f' can be extended to a satisfying assignment of f by assigning some value to x.

Note that the third condition is trivially true for UE-CSP. These techniques and the observation that w.u.p.p. the $\left(\frac{1}{2}-\epsilon\right)n$ 2-clauses do not induce a cycle reduce Lemma 12 to the following lemma.

Lemma 13. *For any $\Delta, \epsilon > 0$, consider a random UE-CSP problem $\mathcal{C}$ on n variables with Δn 3-clauses and $(\frac{1}{2}-\epsilon)n$ 2-clauses where every such formula is equally likely. If $\mathcal{C}$ has no cycle in the 2-clauses, then a.s.:*

(a) every subformula on at most αn variables is satisfiable, and

(b) every subformula on v variables where $\frac{1}{2}\alpha n \leq v \leq \alpha n$ has at least ζn variables of degree ≤ 1.

This proof of this lemma closely follows a similar one from [25].

Proof. Consider any $((2+p),d)$-UE-CSP problem $\mathcal{C}$ and its underlying hypergraph H. A *pendant path* of H is a path of 2-edges whose internal vertices each have degree 2 and do not lie in any 3-edge of H. Trivially, a single vertex is a pendant path of length 0.

For any $r \geq 1$, a Y_r *configuration* consists of:

- r pendant paths and
- a collection of t_2 additional 2-edges and t_3 additional 3-edges whose vertices are all endpoints of the r pendant paths for some t_2, t_3 with $\frac{3}{2}t_2 + 3t_3 \geq \frac{2}{3}r_0 + \frac{5}{3}r_1$

where r_0 is the number of pendant paths of length 0 and $r_1 = r - r_0$.

Let $\mathcal{P}$ be a set of r pendant paths of H such that (i) every vertex of H appears on exactly one path and (ii) $\mathcal{P}$ is minimal in the sense that it is impossible to form a collection of $r-1$ paths satisfying (i) by adding a 2-edge from H to $\mathcal{P}$.

If C' is a minimally unsatisfiable subformula of $\mathcal{C}$, then C' must be connected and have no vertices of degree ≤ 1. By Lemma 14, C' must have a Y_r configuration for some $r \geq 1$. By Lemma 15, there a.s. can not be a Y_r configuration on at most αn variables so $\mathcal{C}$ a.s. has no unsatisfiable formula on at most αn variables.

Consider any subformula F on v variables where $\frac{1}{2}\alpha n \leq v \leq \alpha n$. Since a.s. every such formula does not have a Y_r configuration, then a.s. for each such F, there exists an $r \geq 1$ such that F has at least $\frac{r}{3}$ variables of degree ≤ 1 and a collection of r pendant paths which contain all its variables.

Now, we show that r must be $\Theta(n)$. Let G be the underlying hypergraph of F. By Lemma 16, G has a.s. at most $2ne^2\theta^{-\theta}$ pendant paths of length θ. Since any path of length more than θ contains a path of exactly θ, G has a.s. at most $2ne^2\theta^{-\theta+1}$ vertices on pendant paths of length at least θ. Thus, there exists $\pi > 0$ such that for all $\theta > 3$, G has at most $ne^{-\pi\theta}$ vertices on pendant paths of length at least θ. Pick θ so that $e^{-\pi\theta} < \frac{\alpha}{4}$. Thus, at least $\frac{\alpha}{4}n$ variables of G lie on paths in $\mathcal{P}$ of length less that θ. Therefore, $r > \frac{\alpha}{4\theta}n$ and G has at least ζn vertices of degree ≤ 1 for $\zeta = \frac{\alpha}{12\theta}$.

The following two lemmas closely follow lemmas from [25].

Lemma 14. *If H has at most $\frac{r}{3}$ vertices of degree ≤ 1 and no cycles in the 2-edges then H has a Y_r configuration.*

Proof. Let $\mathcal{P}$ be a minimal set of pendant paths of H such that every vertex of H appears on exactly one path. Let r be the number of paths in $\mathcal{P}$, let r_0 be the number of paths of length 0, and let $r_1 = r - r_0$.

We call the edges of $\mathcal{P}$ *path edges* and the other edges of H *non-path edges*. Note that every non-path edge contains only vertices that are endpoints of the paths in $\mathcal{P}$. Let t_2 be the non-path 2-edges, and let t_3 be the (non-path) 3-edges. We will prove H has a Y_r configuration by proving $\frac{3}{2}t_2 + 3t_3 \geq \frac{2}{3}r_0 + \frac{5}{3}r_1$.

We define a set X to contain exactly those vertices which are an endpoint of a path of $\mathcal{P}$. Thus, $|X| = 2r_1 + r_0$. We form a graph G with vertex set X, and the edges of G are the 2-edges of H which do not lie on a path of $\mathcal{P}$. Note that $t_2 = |E(G)|$.

Let l_1 be the number of components of G with exactly one vertex, and let l_2 be the number of components with exactly two vertices. The remaining components have size at least 3, and thus these components contain $|X| - l_1 - 2l_2$ vertices and at least $\frac{2}{3}(|X| - l_1 - 2l_2)$ edges. Therefore, $t_2 \geq l_2 + \frac{2}{3}(|X| - l_1 - 2l_2)$ and rearranging gives $\frac{3}{2}t_2 + l_1 + \frac{1}{2}l_2 \geq |X| = r_0 + 2r_1$.

Now note that every vertex which had degree 0 in G must either be in a 3-edge or have degree at most 1 in H. Also note that every component of G which has size 2 must have at least 1 vertex which is either in a 3-edge or has degree at most 1 in H. Otherwise, the two vertices are either the two endpoints of the same path in $\mathcal{P}$ which would form a cycle in the 2-edges of H, or endpoints of different paths in $\mathcal{P}$ which would violate the minimality of $\mathcal{P}$. This yields $l_1 + l_2 \leq 3t_3 + s$ where s is the number of vertices of degree at most 1 in H. Thus,

$$r_0 + 2r_1 \leq \frac{3}{2}t_2 + l_1 + \frac{1}{2}l_2 \leq \frac{3}{2}t_2 + l_1 + l_2 \leq \frac{3}{2}t_2 + 3t_3 + s.$$

Since $s \leq \frac{r}{3}$, H has a Y_r configuration. □

Lemma 15. *For any $\Delta, \epsilon > 0$, consider a random hypergraph H on n vertices with Δn 3-edges and $(\frac{1}{2} - \epsilon)n$ 2-edges where every such graph is equally likely. There is some constant $\alpha > 0$ such that a.s. H has no Y_r configuration for any $r < \alpha n$.*

Proof. Fix an $r < \alpha n$ and compute the expected number of Y_r configurations.

Consider any list of 2-edges $e_1, \ldots, e_k$. The probability that they all appear in H is

$$\begin{aligned}
\frac{\binom{\binom{n}{2}-k}{(\frac{1}{2}-\epsilon)n-k}}{\binom{\binom{n}{2}}{(\frac{1}{2}-\epsilon)n}} &= \frac{\left(\binom{n}{2}-k\right)!\left(\left(\frac{1}{2}-\epsilon\right)n\right)!}{\binom{n}{2}!\left(\left(\frac{1}{2}-e\right)n-k\right)!} \\
&= \left(\frac{\left(\frac{1}{2}-\epsilon\right)n}{\binom{n}{2}}\right)\cdots\left(\frac{\left(\frac{1}{2}-\epsilon\right)n-k}{\binom{n}{2}-k}\right) \\
&\leq \left(\frac{\left(\frac{1}{2}-\epsilon\right)n}{\binom{n}{2}}\right)^k \\
&= \left(\frac{2\left(\frac{1}{2}-\epsilon\right)}{n-1}\right)^k \\
&< \left(\frac{2\left(\frac{1}{2}-\epsilon'\right)}{n}\right)^k
\end{aligned}$$

for some $0 < \epsilon' < \epsilon$.

As before, we let X be the set of vertices where are endpoints of the pendant paths of the Y_r configuration. There are at most $\binom{n}{r}n^{r_1}$ choices for the endpoints of the r paths. Suppose the number of 2-edges in the paths are $l_1, \ldots, l_r$ and let $L = l_1 + \cdots + l_r$. Then there are n^{L-r} choices for the interior vertices of the paths. We multiply by the probability that all L of these edges appear and that there are t_2 other 2-edges and t_3 3-edges on the endpoints. First, assume that t_2 and t_3 are both at least $\frac{r}{100}$. This gives an upper bound of

$$\begin{aligned}
&\sum_{l_1,\ldots,l_r\geq 0}\binom{n}{r}n^{r_1}n^{L-r}\left(\frac{2(\frac{1}{2}-\epsilon')}{n}\right)^L\binom{(\frac{1}{2}-\epsilon)n}{t_2}\binom{\Delta n}{t_3}\left(\frac{|X|}{n}\right)^{2t_2+3t_3} \\
&\leq \left(\frac{ne}{r}\right)^r n^{r_1-r}\left(\frac{\frac{1}{2}ne}{t_2}\right)^{t_2}\left(\frac{\Delta ne}{t_3}\right)^{t_3}\left(\frac{2r}{n}\right)^{2t_2+3t_3}\sum_{l_1,\ldots,l_r}(1-2\epsilon')^L \\
&\leq \left(\frac{r}{n}\right)^{t_2+2t_3-r_1} e^{t_2+t_3+r}\Delta^{t_3}2^{t_2+3t_3}100^{t_2+t_3}\left(\sum_{l\geq 0}(1-2\epsilon')^l\right)^r \qquad (4) \\
&\leq \left(\frac{\gamma_1 r}{n}\right)^{t_2+2t_3-r_1} \qquad (5) \\
&\leq \left(\frac{\gamma_1 r}{n}\right)^{r/9} \qquad (6)
\end{aligned}$$

for some $\gamma_1 > 0$. Inequality (4) follows because $t_2, t_3 \geq \frac{r}{100}$, (5) follows because $t_2+2t_3-\frac{2}{3}r_0-\frac{10}{9}r_1 \geq 0$, and (6) follows because $t_2+2t_3-r_1 = \frac{2}{3}(\frac{3}{2}t_2+3t_3)-r_1 \geq \frac{2}{3}(\frac{2}{3}r_0+\frac{5}{3}r_1)-r_1 \geq \frac{2}{3}(\frac{5}{3}r_1)-r_1 = \frac{r}{9}$.

If $t_2 \leq \frac{r}{100}$ then $t_3 \geq (\frac{2}{9}-\frac{1}{200})r_0+(\frac{5}{9}-\frac{1}{200})r_1$. For such a t_2, compute the expected number of collections of r pendant paths along with t_3 3-clauses on their endpoints. As above, the expected number is upper bounded with:

$$\left(\frac{ne}{r}\right)^r n^{r_1-r}\left(\frac{e\Delta n}{t_3}\right)^{t_3}\left(\frac{|X|}{n}\right)^{3t_3}\left(\sum_{l\geq 0}(1-2\epsilon')^l\right)^r < \left(\frac{\gamma_2 r}{n}\right)^{r/10}$$

for some $\gamma_2 > 0$.

If $t_3 \leq \frac{r}{100}$ then $t_2 \geq (\frac{4}{9}-\frac{2}{100})r_0+(\frac{10}{9}-\frac{2}{100})r_1$. For such a t_3, and similar to above, the expected number of collections of r pendant paths and t_2 2-edges on the endpoints is upper bounded by:

$$\left(\frac{ne}{r}\right)^r n^{r_1-r}\left(\frac{\frac{1}{2}ne}{t_2}\right)^{t_2}\left(\frac{|X|}{n}\right)^{2t_2}\left(\sum_{l\geq 0}(1-2\epsilon')^l\right)^r < \left(\frac{\gamma_3 r}{n}\right)^{r/11}$$

for some $\gamma_3 > 0$.

Let $\gamma = \max\{\gamma_1, \gamma_2, \gamma_3\}$. As there are $O(r)$ choices for t_2, t_3, it suffices to show that

$$\sum_{r=1}^{\alpha n} r \left(\frac{\gamma r}{n}\right)^{r/11} = o(1).$$

The first $\log n$ terms of this sum add up to at most $O\left(\frac{\log n}{n^{1/11}}\right)$ and if $\alpha < \frac{1}{2\gamma}$ then the rest add up to at most $\sum_{i \geq \log n} i \left(\frac{1}{2}\right)^{i/11} = o(1)$. □

The proof of this final lemma is an exercise in the second moment method on random graphs.

Lemma 16. *For any $\epsilon > 0$ and any $c > 0$, a uniformly random graph with n vertices and $\left(\frac{1}{2} - \epsilon\right) n$ edges has a.s. at most $(1+c)ne^2\theta^{-\theta}$ pendant paths of length θ.*

Proof. Using a well known property of random graphs, we can consider a model with n vertices and each of the $\binom{n}{2}$ edges existing independently with probability $p = \frac{1}{n}$.

Let X be the number of pendant paths of length θ. The expected value of X is bounded above by the number of choices for the θ vertices, the probability that each edge on the path exists, the probability there is no edge from the interior path vertices to the rest of the graph, and the probability the path is induced:

$$\begin{aligned}
\mathbf{Exp}(X) &\leq \binom{n}{\theta} p^{\theta-1} (1-p)^{(n-\theta)(\theta-2)} (1-p)^{\binom{\theta-1}{2}} \\
&\sim \left(\frac{ne}{\theta}\right)^{\theta} \left(\frac{1}{n}\right)^{\theta-1} e^{\theta+2} \\
&= ne^2\theta^{-\theta}.
\end{aligned}$$

Using the second moment method, we show the expected number is highly concentrated about its mean by summing over all sets of θ vertices which intersect with a given path multiplied by the probability that the intersecting set is also a pendant path. In the calculations below, k is the number of the θ vertices which intersect with the given path.

$$\begin{aligned}
\mathbf{Exp}(X^2) = \mathbf{Exp}(X) \Bigg[& 1 + \sum_{k=1}^{\theta-1} 2 \binom{n-\theta}{k} p^k (1-p)^{(n-\theta-k+1)k-1} (1-p)^{\binom{k}{2}} \\
& + \binom{n-\theta}{\theta} p^{\theta-1} (1-p)^{(n-2\theta+2)(\theta-2)} (1-p)^{\binom{\theta-1}{2}} \Bigg] \\
\sim \mathbf{Exp}(X) & \left[1 + \sum_{k=1}^{\theta-1} 2 \left(\frac{ne}{k}\right)^k \left(\frac{1}{n}\right)^k e^{-k} + \mathbf{Exp}(X) \right] \\
\sim \mathbf{Exp}(X)^2 & (1 + o(1)).
\end{aligned}$$

So by Chebyshev's Inequality, the probability that, for any $c > 0$, $X > (1+c)\mathbf{Exp}(X)$ is $o(1)$. □

References

1. Selman, B., Mitchell, D.G., Levesque, H.J.: Generating hard satisfiability problems. Artificial Intelligence **81** (1996) 17–29
2. Kirkpatrick, S., Selman, B.: Critical behavior in the satisfiability of random boolean expressions. Science **264** (1994) 1297–1232
3. Hajiaghayi, M.T., Sorkin, G.B.: The satisfiability threshold of random 3-SAT is at least 3.52. arxiv.org/abs/math.CO/0310193 (2003)
4. Kaporis, A.C., Kirousis, L.M., Lalas, E.G.: Selecting complementary pairs of literals. In: Electronic Notes in Discrete Mathematics. Volume 16., Elsevier (2003)
5. Dubois, O., Boufkhad, Y., Mandler, J.: Typical random 3-SAT formulae and the satisfiability threshold. Technical Report TR03-007, Electronic Colloquium on Computational Complexity (2003)

6. Friedgut, E.: Sharp thresholds of graph properties, and the k-SAT problem. Journal of the American Mathematical Society **12** (1999) 1017–1054
7. Chvátal, V., Reed, B.: Mick gets some (the odds are on his side). In: Proceedings of the 33rd Annual Symposium on Foundations of Computer Science. (1992) 620–627
8. Goerdt, A.: A threshold for unsatisfiability. Journal of Computer and System Sciences **53** (1996) 469–486
9. Fernandez de la Vega, W.: On random 2-SAT. Manuscript (1992)
10. Monasson, R., Zecchina, R., Kirkpatrick, S., Selman, B., Troyansky, L.: Phase transitions and search cost in the 2 + p-sat problem. In: 4th Workshop on Physics and Computation. (1996)
11. Achlioptas, D., Kirousis, L.M., Kranakis, E., Krizanc, D.: Rigourous results for random $(2 + p)$-SAT. Theoretical Computer Science **265** (2001) 109–129
12. Achlioptas, D., Beame, P., Molloy, M.: A shart threshold in proof complexity yields lower bounds for satisfiability search. In: Proceedings of the 33rd Annual ACM Symposium on Theory of Computing. (2001) 337–346
13. Chao, M.T., Franco, J.: Probibalistic analysis of two heuristics for the 3-satisfiability problem. SIAM Journal of Computing **15** (1986) 1106–1118
14. Frieze, A., Suen, S.: Analysis of two simple heuristics on a random instance of k-SAT. Journal of Algorithms **20** (1996) 312–355
15. Cocco, S., Monasson, R., Montanari, A., Semerjian, G.: Approximate analysis of search algorithms with "physical" methods. arxiv.org/abs/cs.CC/0302003 (2003)
16. Martin, O.C., Monasson, R., Zecchina, R.: Statistical mechanics methods and phase transitions in optimization problems. Theoretical Computer Science **265** (2001) 3–67
17. Mézard, M., Ricci-Tersenghi, F., Zecchina, R.: Alternative solutions to diluted p-spin models and XORSAT problems. Journal of Statistical Physics **111** (2003) 505–533
18. Dubois, O., Mandler, J.: The 3-XORSAT threshold. In: Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science. (2002) 769–778
19. Connamacher, H., Molloy, M.: The exact satisfiability threshold for a potentially intractable random constraint satisfaction problem. In Preparation (2003)
20. Achlioptas, D.: A survey of lower bounds for random 3-SAT via differential equations. Theoretical Computer Science **265** (2001) 159–185
21. Wormald, N.: Differential equations for random processes and random graphs. Annals of Applied Probability **5** (1995) 1217–1235
22. Chao, M.T., Franco, J.: Probabilistic analysis of a generalization of the unit clause literal selection heuristic for the k-satisfiability problem. Information Science **51** (1990) 289–314
23. Ben-Sasson, E., Wigderson, A.: Short proofs are narrow - resolution made simple. Journal of the ACM **48** (2001) 149–169
24. Mitchell, D.: Resolution complexity of random constraints. In: Principles and Practices of Constraint Programming – CP 2002. (2002) 295–309
25. Molloy, M., Salavatipour, M.: The resolution complexity of random constraint satisfaction problems. Preprint. Extended abstract in the Proceedings of FOCS 2003, 330-339 (2003)