A Sharp Threshold in Proof Complexity

Dimitris Achlioptas
Microsoft Research
One Microsoft Way
Redmond, WA 98052
optas@microsoft.com

Paul Beame[*]
Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350
beame@cs.washington.edu

Michael Molloy[†]
Department of Computer Science
University of Toronto
Toronto, Ontario M5S 1A4
molloy@cs.toronto.edu

ABSTRACT

We give the first example of a sharp threshold in proof complexity. More precisely, we show that for any sufficiently small $\epsilon > 0$ and $\Delta > 2.28$, random formulas consisting of $(1-\epsilon)n$ 2-clauses and Δn 3-clauses, which are known to be unsatisfiable almost certainly, almost certainly require resolution and Davis-Putnam proofs of unsatisfiability of exponential size, whereas it is easily seen that random formulas with $(1+\epsilon)n$ 2-clauses (and Δn 3-clauses) have linear size proofs of unsatisfiability almost certainly.

A consequence of our result also yields the first proof that typical random 3-CNF formulas at ratios below the generally accepted range of the satisfiability threshold (and thus expected to be satisfiable almost certainly) cause natural Davis-Putnam algorithms to take exponential time to find satisfying assignments.

1. INTRODUCTION

The satisfiability problem has received a great deal of study as the canonical NP-complete problem. In the last several years the very universality and flexibility of the satisfiability problem that made it a natural starting point for NP-completeness has also made it the basis for significant progress in the solution of a variety of practical problems including problems in constraint satisfaction [31], planning [24, 23], and symbolic model checking [9]. The basic tools for these advances are some very tight and efficient implementations of satisfiability algorithms using backtracking search based on the Davis-Putnam/DLL (DPLL) procedure [16, 15] and using heuristic search based on hill-climbing and random walks [31, 30]. In a sense, these satisfiability algorithms have become the hammer and there is now a small industry turning computational problems into nails.

A folklore property of these satisfiability algorithms is that they work extremely well at handling typical satisfiable formulas, although outlying satisfiable formulas can cause very bad behavior for any specific algorithm. (One highly successful strategy is to use randomized DPLL algorithms and re-start them with different random bits if they begin to take too long [21, 20].)

It has previously been shown that DPLL algorithms are exponentially inefficient for proving unsatisfiability for typical unsatisfiable formulas [13, 8, 6, 7]. We give a method to prove the first lower bounds on the running times of DPLL algorithms for typical *satisfiable* CNF formulas. We show that certain natural DPLL algorithms when applied to random 3-CNF formulas for which there is strong empirical evidence of satisfiability (density below 3.98) almost certainly generate subproblems consisting of unsatisfiable mixed formulas of 2- and 3-clauses for which we can prove that exponential-size proofs of unsatisfiability are required. The key to this result is the proof complexity lower bound for these mixed formulas.

Let $\mathcal{F}^n_{\epsilon,\Delta}$ be the distribution of random CNF formulas with $(1-\epsilon)n$ 2-clauses and Δn 3-clauses, for some arbitrary *constants* $\Delta, \epsilon > 0$. Let $res(F)$ and $DLL(F)$ be the sizes of the minimal resolution and Davis-Putnam/DLL proofs of the unsatisfiability of F. We will say that a sequence of random events $\mathcal{E}_n$ occurs with high probability (w.h.p.) if $\lim_{n\to\infty} \Pr[\mathcal{E}_n] = 1$ and with constant probability if $\liminf_{n\to\infty} \Pr[\mathcal{E}_n] > 0$.

THEOREM 1. *For every $\Delta, \epsilon > 0$, if $F \sim \mathcal{F}^n_{\epsilon,\Delta}$, then w.h.p., $res(F) = 2^{\Omega(n)}$ and $DLL(F) = 2^{\Omega(n)}$.*

This bound is interesting in its own right since it yields the first example of a sharp threshold for the proof complexity of unsatisfiable formulas. More precisely, for $\Delta > 2.28$ formulas from $\mathcal{F}^n_{\epsilon,\Delta}$ are unsatisfiable almost certainly [4] and require exponential-size resolution and DPLL proofs of this fact, but if we slightly increase the number of 2-clauses from $(1-\epsilon)n$ to $(1+\epsilon)n$, the 2-CNF subformula alone becomes unsatisfiable almost certainly and this unsatisfiability problem becomes trivial for resolution or DPLL proofs.

Mixed formulas consisting of 2- and 3-clauses arise for two natural reasons. A frequent observation about converting problems from other domains into satisfiability problems is that they typically become mixed CNF formulas with a substantial number of clauses of length 2 along with clauses of length 3. Also, as DPLL algorithms run, they recursively solve satisfiability on restricted versions of their input CNF formula which are mixtures of clauses of length at least 2. Randomly-chosen k-CNF formulas are an important test case for satisfiability algorithms. When given randomly-chosen 3-CNF formulas as input, many DPLL algorithms produce restricted mixed formulas that are distributed precisely in the form that we analyze.

A fundamental conjecture about random k-CNF formulas says that for each $k \geq 2$, there is a constant α_k, the *satisfiability threshold,* such that a random k-CNF formula of clause-variable ratio Δ is almost certainly satisfiable for $\Delta < \alpha_k$ and almost certainly unsatisfiable if $\Delta > \alpha_k$. It is known that 1 is the threshold for

[*] Research supported by NSF grant CCR-9800124.
[†] Research supported by NSERC and by a Sloan Fellowship

STOC'01, July 6-8, 2001, Hersonissos, Crete, Greece.
Copyright 2001 ACM 1-58113-349-9/01/0007 ...$5.00.

random 2-SAT [12, 19] and empirically, it appears that there is a threshold for random 3-SAT around 4.2 [32, 25]. The best proven bounds show that random 3-SAT formulas are almost certainly satisfiable for $\Delta < 3.26$ [5] and almost certainly unsatisfiable for $\Delta > 4.598$ [22] and it is known that the 'phase' transition from satisfiable to unsatisfiable formulas is asymptotically sharp [17]. The proven lower bounds on the satisfiability thresholds have all been derived by analyzing specific DPLL-like algorithms without backtracking ('card-type/myopic algorithms' in the terminology of [1, 5]). The formulas we consider are at densities bounded well below the empirical 4.2 threshold and thus should be almost certainly satisfiable.

An extension of random 3-SAT problems to mixed formulas has led to the study of the so-called $(2+p)$-SAT problem. Here one looks at randomly-generated formulas on n variables with a mix of clauses of length 2 and 3 where the fraction of clauses of length 3 is p. Using a mixture of empirical results and heuristic, non-rigorous techniques of statistical physics, Kirkpatrick et. al. [27, 28, 29] gave evidence that one could add up to roughly $2n/3$ random 3-clauses (corresponding to p around 0.4) and have no impact on typical satisfiability problems in that the point at which the threshold is reached is dependent solely on the number of 2-clauses. Furthermore they suggested that the phase transition for the $(2+p)$-SAT problem itself changed character from a so-called first-order transition (discontinuous "order parameter") representative of 2-SAT to a second-order transition (continuous "order parameter") believed to be representative of 3-SAT at this point and suggested that this second-order phase transition somehow characterizes NP-hard problems. (This latter conjecture has been put in further perspective by the results in [3].)

In [4], Achlioptas et. al. proved a number of rigorous results about $(2+p)$-SAT. They showed that adding $2n/3$ randomly chosen 3-clauses to a random formula with $(1-\epsilon)n$ 2-clauses yields a formula that is almost surely satisfiable (and this assignment can be found by a simple card-type algorithm), while adding $2.28n$ random 3-clauses yields an almost certainly unsatisfiable formula. Moreover, they proved that the transition from satisfiability to unsatisfiability is sharp and empirical evidence suggests that the $2n/3$ bound is much closer to the truth [29]. Furthermore $2n/3$ is the provable limit for showing satisfiability by card-type algorithms [1] and, in [2], it was conjectured that it is tight for all algorithms.

The card-type algorithms analyzed for 3-SAT have the property that they make irrevocable choices to the partial assignment and at each point in their execution the restricted formula that remains is an unbiased random formula characterized by a pair of integers (c_2n, c_3n) describing the number of 2- and 3-clauses respectively that remain. The algorithms will succeed almost certainly if they exhaust the 3-clauses without c_2 ever reaching 1. In fact, if they reach $((1-\epsilon)n, 2n/3)$ without c_2 ever reaching 1 they will succeed with constant probability [4].

One can extend the card-type algorithms, such as UC (unit clauses first) and GUC [11], into full backtracking DPLL algorithms in a variety of different ways so that the execution of the original algorithm is the first path explored in the tree of recursive calls. If the original card-type algorithm reaches $((1-\epsilon)n, c_3n)$ where c_3 is at least 2.28, the resulting formula is almost certainly unsatisfiable, and by our argument it almost certainly requires an exponential-size DPLL proof. Thus, once this node has been explored in the backtracking search, the search cannot leave the sub-tree for an exponentially long time.

Our results show that this happens with constant probability for UC started $3.81n$ 3-clauses and for GUC started with $3.98n$ 3-clauses (it would happen almost certainly except that the card-type algorithm might reach a dead end before then) and thus with constant probability *any* backtracking extension of UC and GUC applied to random 3-CNF formulas above these ratios will run for exponential time. Furthermore, by generalizing a limited backtracking heuristic used by Frieze and Suen [18] into a full backtracking heuristic we create natural DPLL algorithms, UC-FS and GUC-FS, extending UC and GUC respectively, for which we can show that above exponential lower bounds hold almost certainly rather than just with constant probability.

These results shed light on a widely-cited and repeated observation of Selman, Mitchell, and Levesque [32], based on experiments with ORDERED-DLL, a backtracking version of an algorithm probabilistically equivalent to UC, on small problems, was that random 3-SAT is easy in the satisfiable region up to the 4.2 threshold, becomes sharply much harder at the threshold and quickly becomes easy again at larger densities in the unsatisfiable region. The upper end of this 'easy-hard-easy' characterization is somewhat misleading since it is known that at any constant ratio above the threshold any DPLL or resolution algorithm almost certainly requires exponential size proofs of unsatisfiability [13]. By now the rate of decline in proof complexity as the ratio is increased has been analyzed as well [6].

Our new results show that the lower end of this characterization is also somewhat misleading; in fact, our results show that the exponentially hard region for ORDERED-DLL begins at least at ratio 3.81, well before ratio 4.2. (This concurs with recent experimental evidence that even the best of current DPLL implementations seem to have bad behavior below the threshold [14].) As we discuss in section 7, our upper bounds on the number of 3-clauses needed to cause exponential behavior in satisfiability algorithms will be readily improved with any improvement on the upper bound for unsatisfiability in random $(2+p)$-SAT. In fact if it turns out that, as was conjectured in [2], for every $\delta > 0$ there exists $\epsilon > 0$ such that a random formula with $(2/3+\delta)n$ 3-clauses and $(1-\epsilon)n$ 2-clause is unsatisfiable w.h.p. then our results would imply a perfectly sharp characterization for these DPLL algorithms.

Our lower bound on the proof complexity of mixed formulas is similar in general spirit to other lower bounds for resolution algorithms but requires considerably more subtlety. We first prove a number of detailed combinatorial properties of random 2-CNF formulas with $(1-\epsilon)n$ clauses. To do this we consider the standard directed graphs associated with 2-CNF formulas and, for such graphs, we introduce the notion of the clan of a vertex. Clans seem to be the appropriate extension of "connected components" in this context, allowing for an amortization of the boundary of the 2-CNF formula. By carefully bounding the number of vertices in clans of each size we show that random 2-CNF with $(1-\epsilon)n$ clauses, w.h.p. have properties that guarante that almost all extensions by linear-sized 3-CNF formulas yield exponential size resolution (and DPLL) proofs. The latter argument relies on specialized sharp moment bounds as well particular properties of clans.

2. BOUNDING RESOLUTION REFUTATION SIZE

The resolution rule allows one to derive a clause $(A \vee B)$ from two clauses $(A \vee x)$ and $(B \vee \bar{x})$. A resolution refutation of an unsatisfiable CNF formula F begins with the clauses of F and by a sequence of inferences using the resolution rule derives the empty clause. The size of a resolution refutation is the number of clauses appearing in the proof. Given F, let $res(F)$ be the length of the shortest resolution refutation of F. The Davis-Putnam/DLL algorithm on a CNF formula F performs a backtracking search for a satisfying assignment of F by extending partial assignments until they either reach a satisfying assignment or violate a clause of F. It is well known that for an unsatisfiable formula F, the tree of nodes explored by any DLL algorithm can be converted to a resolution refutation of F where the pattern of inferences forms a tree. Let $DLL(F)$ be the size of the smallest such refutation, i.e. the size of the smallest DLL tree associated with F.

DEFINITION 2. *Let F be an an arbitrary CNF formula.*

- *Given a set F of clauses, a literal x is* pure *in F if and only if x appears in F but $\neg x$ does not appear in F.*
- *$V(F)$ denotes the set of variables of F.*
- *$|F|$ denotes the number of clauses in F.*
- *The* degree *of a variable v, $\deg(v)$, is the number of clauses of F containing one of $v, \bar{v}$ (analogously for literals). The* average degree *of F is $\left(\sum_v \deg(v)\right)/|V(F)|$.*
- *With any 2-CNF formula F on variables $\{x_1, \ldots, x_n\}$ we associate the following directed graph $\vec{D}(F)$:*
  - *The vertex set is $\{x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$.*
  - *The edge set is*

    $$\{(\bar{x} \to y), (\bar{y} \to x) : (x \vee y) \text{ is a clause in } F\} \ .$$
- *We say that a literal ℓ is* near-pure *in F if $\deg(\bar{\ell}) = 1$.*
- *We say that a directed cycle $C = \ell_1 \to \ell_2 \to \cdots \to \ell_q \to \ell_1$ in $\vec{D}(F_2)$ is* pure *if all of $\ell_1, \ldots, \ell_q$ are near pure in F. (Note that each literal can appear in at most one pure cycle.)*
- *F is μ-*pure *if it has at most $\mu|V(F)|$ pure items, i.e. pure literals and cycles.*

The *boundary* of F, $b(F)$, is the set of pure items in F. Note that this generalizes the definition of boundary from [8] where the boundary was defined to be those literals whose variables appeared in precisely one clause of F. Define the *satisfiability threshold for F*, $s(F)$, to be the size of the minimal unsatisfiable subset of G.

Given a set of clauses F we say that $F \Rightarrow_{Res} C$ if and only if there is a resolution derivation of C from F that uses all clauses in F. Define the *sub-critical expansion of F*, $e(F) = \max_{s \leq s(F)} \min\{|b(G)| : s/2 \leq |G| < s\}$. The following are extensions of [8].

LEMMA 3. *If $F \Rightarrow_{Res} C$ then C contains at least $|b(F)|$ literals.*

PROOF. This is trivial for pure literals since, once they appear in the derivation, they cannot be removed. For pure cycles, observe that once a literal from the pure cycle appears in a clause in the derivation, all clauses derived from it will also contain a literal from that cycle. The argument is by induction on the proof in topological order: Suppose that a literal x from the cycle appears in clause A and does not appear in clause D where D is derived from A and some clause B by the resolution rule. Therefore x must appear negatively in B and the resolution rule must use x. The only way that $\bar{x}$ can be in B is if B is derived using the one negative clause $\bar{x} \vee y$ that appears in the pure cycle containing x. Therefore by the induction hypothesis B contains some literal in this pure cycle and this literal will appear in D. Since each literal is in at most one pure cycle, the lemma follows. □

Using Lemma 3 the following is immediate from [8].

PROPOSITION 4. *If all clauses in a formula F have size at most k, then $res(F) = 2^{\Omega([e(F)-k]^2/n)}$ and $DLL(F) \geq 2^{e(F)-k}$.*

For each n, let us fix a canonical set of n variables and for each fixed $k \geq 0$, let $C_k(n)$ denote the set containing all $2^k\binom{n}{k}$ non-trivial k-clauses on our canonical set of variables. We will consider a random formula F on n variables formed by selecting uniformly, independently and with replacement $m_2 = m_2(n)$ clauses from $C_2(n)$ and $m_3 = m_3(n)$ clauses from $C_3(n)$.

Let $\mathcal{F}^n_{\epsilon,\Delta}$ be the distribution where $m_2 = (1-\epsilon)n$ and $m_3 = \Delta n$, for some arbitrary *constants* $\Delta, \epsilon > 0$.

LEMMA 5. *For every $\Delta, \epsilon > 0$ there exist $\zeta = \zeta(\Delta, \epsilon) > 0$ and $\mu = \mu(\Delta, \epsilon) > 0$ such that*

1. *W.h.p. F contains no unsatisfiable subformula on $v \leq \zeta n$ variables.*
2. *W.h.p. F contains no μ-pure subformula on $\frac{1}{2}\zeta n \leq v \leq \zeta n$ variables.*

COROLLARY 6. *For $F \sim \mathcal{F}^n_{\epsilon,\Delta}$, w.h.p., $res(F) = 2^{\Omega(n)}$ and $DLL(F) = 2^{\Omega(n)}$.*

The overall proof strategy will be to analyze almost certain properties of the 2-clauses of F and show that such subformulas must have large boundaries and then to show that the addition of 3-clauses does not significantly reduce the size of the boundary. (Since the 2-clause subformula of F alone is almost certainly satisfiable, the fact that the two clauses alone have a large boundary does not contradict the fact that 2-SAT is easy for resolution.) This argument is subtle because the 2-clauses of F are so close to being unsatisfiable themselves and because we need to handle all possible subformulas among the 2-clauses. The latter requirement necessitates the introduction of a somewhat unusual graph-theoretic concept in the di-graph associated with the 2-clauses of F.

3. BOUNDING PURITY

By Lemma 3, any minimal unsatisfiable formula must have no pure items. Therefore, it will be convenient to restate Lemma 5 in terms of the following definition.

DEFINITION 7. A subformula H of F on v variables is (μ, ζ)-*pure* if: i) $v < \frac{1}{2}\zeta n$ and H is 0-pure, or, ii) $\frac{1}{2}\zeta n \leq v \leq \zeta n$ and H is μ-pure.

Hence, Lemma 5 can be restated as

LEMMA 8. *For every $\Delta, \epsilon > 0$, there exists $\zeta = \zeta(\Delta, \epsilon) > 0$, $\mu = \mu(\Delta, \epsilon)$ such that w.h.p. F contains no (μ, ζ)-pure subformula.*

To prove Lemma 5 we will need need to consider two possibilities for the fraction of 3-clauses appearing in a potential (μ, ζ)-pure subformula. In particular, it will be relatively easy to prove that w.h.p. every subformula rich in 3-clauses is not (μ, ζ)-pure (Lemma 11). Proving the same assertion for subformulas where the fraction of 3-clauses is arbitrarily small (Lemma 10) will be harder, comprising Sections 4 and 5. More precisely, let us say that

DEFINITION 9. *A formula F is δ-*rich *if it has at least $\delta|V(F)|$ 3-clauses, and δ-*poor *otherwise.*

Lemma 8 will follow readily from the following two lemmas.

LEMMA 10. *For every $\Delta, \epsilon > 0$ and all $\zeta \leq \zeta_1(\Delta, \epsilon)$, there exist $\delta_1 = \delta_1(\Delta, \epsilon, \zeta) > 0$ and $\mu_1 = \mu_1(\Delta, \epsilon, \zeta)$ such that w.h.p. $\mathcal{F}^n_{\epsilon,\Delta}$ contains no δ_1-poor, (μ_1, ζ)-pure subformula.*

LEMMA 11. *For every $\Delta, \epsilon, \delta > 0$, and all $\zeta \leq \zeta_2(\Delta, \epsilon, \delta)$, there exists $\mu_2 = \mu_2(\Delta, \epsilon, \zeta)$ such that w.h.p. $\mathcal{F}^n_{\epsilon,\Delta}$ contains no δ-rich, (μ_2, ζ)-pure subformula.*

PROOF OF LEMMA 8. Given $\Delta, \epsilon > 0$ we start by applying Lemma 10 with $\zeta = \zeta_1(\Delta, \epsilon)$. This determines a value $\delta^* = \delta_1(\Delta, \epsilon, \zeta_1) > 0$. We then apply Lemma 11 with $\delta = \delta^*$ and $\zeta = \zeta_2(\Delta, \epsilon, \delta^*) > 0$.

As a result, for $\zeta = \min\{\zeta_1(\Delta, \epsilon), \zeta_2(\Delta, \epsilon, \delta^*)\} > 0$ and $\mu = \max\{\mu_1(\Delta, \epsilon, \zeta), \mu_2(\Delta, \epsilon, \zeta)\} > 0$, we see that w.h.p. there are no δ^*-poor, (μ, ζ)-pure subformulas and no δ^*-rich, (μ, ζ)-pure subformulas. □

To prove Lemma 11 we will need the following

LEMMA 12. *For every $\Delta, \epsilon, \beta > 0$ there is $\zeta = \zeta(\Delta, \epsilon, \beta) > 0$ such that w.h.p. no subformula of $F \sim \mathcal{F}^n_{\epsilon,\Delta}$ on $v \leq \zeta n$ variables has average degree at least $2 - \beta/2$ and at least βv 3-clauses.*

PROOF. For a given $\beta > 0$ let α be such that $2\alpha + 3\beta = 2 - \beta/2$, implying $\alpha + 2\beta - 1 = \beta/4$. We will bound the expected number of subformulas on v variables having αv 2-clauses and βv 3-clauses. Using this bound we will show that there exists $\zeta = \zeta(\Delta, \epsilon, \beta) > 0$ such that the total expected number of such subformulas for $1 \leq v \leq \zeta n$ variables is $o(1)$.

For a fixed v, we can bound the expected number, $Q_\beta(v)$, of such subformulas by

$$\binom{n}{v}\binom{(1-\epsilon)n}{\alpha v}\binom{\Delta n}{\beta v}\left(\frac{v}{n}\right)^{2\alpha v + 3\beta v} \tag{1}$$

$$\begin{aligned}
&\leq \left(\frac{en}{v}\right)^{v}\left(\frac{e(1-\epsilon)n}{\alpha v}\right)^{\alpha v}\left(\frac{e\Delta n}{\beta v}\right)^{\beta v}\left(\frac{v}{n}\right)^{2\alpha v + 3\beta v}\\
&= \left(\frac{(1-\epsilon)^{\alpha} e^{1+\alpha+\beta}\Delta^{\beta}}{\alpha^{\alpha}\beta^{\beta}} \times \left(\frac{v}{n}\right)^{\alpha+2\beta-1}\right)^{v}\\
&= \left(K \times \left(\frac{v}{n}\right)^{\beta/4}\right)^{v} \quad ,
\end{aligned} \tag{2}$$

for some (constant) $K = K(\beta, \Delta, \epsilon)$. It is easy to see that for $\zeta = \zeta(\beta, K)$ sufficiently small and all n sufficiently large, if $v \leq \zeta n$ the right hand side of (2) is decreasing with v. Therefore,

$$\sum_{1 \leq v \leq \zeta n} Q_\beta(v) \leq \sum_{v=1}^{8/\beta} K n^{-\beta/4} + \sum_{v > 8/\beta} K \times \frac{(8/\beta)^{\beta/4}}{n^2} = o(1).$$

□

PROOF OF LEMMA 11. We first observe that any μ-pure subformula H on v variables must contain at most μv pure literals and therefore have average degree at least $2(1-\mu)$. Moreover, if H is δ-rich, it contains at least δv 3-clauses. We will take $\mu_2 = \delta/4$. Thus, any δ-rich, μ_2-pure subformula must have average degree at least $2 - \delta/2$ and contain at least δv 3-clauses. But from Lemma 12, there exists $\zeta_2 = \zeta(\Delta, \epsilon, \delta) > 0$ such that w.h.p. F contains no such subformula on $v \leq \zeta_2 n$ variables. □

To prove Lemma 10 we will in fact prove a stronger lemma. In particular, rather than proving the lemma's assertion for $\mathcal{F}^n_{\epsilon,\Delta}$, we will prove it for an arbitrary formula on n variables formed by starting with a 2-CNF formula satisfying certain properties and adding to it $m_3 = \Delta n$ random 3-clauses. To complete the proof, in appendix A, we prove that F_2 satisfies these properties w.h.p.

4. THE δ-POOR CASE

As mentioned earlier, we will prove the assertion of Lemma 10 for formulas formed by starting with a 2-CNF formula satisfying certain properties and adding m_3 random 3-clauses. To describe these properties we need to introduce the following definitions.

DEFINITION 13. Let F be an arbitrary 2-CNF formula. For literals x, y appearing in F let us write $x \leadsto_F y$ iff $x = y$ or there exists a directed path in $\vec{D}(F)$ from x to y.

- For each literal x we let $\mathrm{In}_F(x) = \{y : y \leadsto_F x\}$.
- For a set of literals S let $G(S) = G_F(S)$ be the undirected graph formed by considering the subgraph of $\vec{D}(F)$ induced by the vertices corresponding to S and ignoring the direction of arcs.
  - We will say that $\mathrm{In}_F(x)$ is *tree-like* if $G(\mathrm{In}_F(x))$ contains *no* cycle.
  - We will say that $\mathrm{In}_F(x)$ is *simple* if $G(\mathrm{In}_F(x))$ contains *at most one* cycle.
- For each literal x in F, the *clan* of x, $\mathrm{Clan}_F(x) = \mathrm{In}_F(x) \cup \bigcup_{y \in \mathrm{In}_F(x)} \mathrm{In}_F(\bar{y})$.
- $T_i(F) = |\{x : |\mathrm{Clan}_F(x)| = i\}|$.

Lemma 10 will follow readily from the following two lemmas.

LEMMA 14. *Fix $\rho \in (0, 1)$ and $\Delta > 0$. Let F^* be a formula formed by taking*

- *Any set of clauses from $C_2(n)$ such that the resulting formula F_2^* satisfies:*
  1. *For every literal ℓ, $\mathrm{In}_{F_2^*}(\ell)$ is simple.*
  2. *There are at most $\log n$ literals ℓ, such that $\mathrm{In}_{F_2^*}(\ell)$ is not tree-like.*
  3. *For all i, $T_i(F_2^*) \leq 2(1-\rho)^i n$.*
- *No more than Δn clauses from $C_3(n)$, chosen uniformly independently and with replacement.*

For all $\zeta \leq \zeta_0(\Delta, \rho)$, there exist $\delta = \delta_0(\Delta, \rho, \zeta) > 0$, and $\mu = \mu_0(\Delta, \rho, \zeta) > 0$ such that, w.h.p. F^* contains no δ-poor, (μ, ζ)-pure subformula.

LEMMA 15. *Fix $\epsilon > 0$ and let F_2 be a random 2-SAT formula formed by selecting uniformly, independently and with replacement $m_2 \leq (1-\epsilon)n$ 2-clauses from $C_2(n)$. There exists $\rho = \rho(\epsilon)$, such that w.h.p. F_2 simultaneously satisfies all three conditions of Lemma 14 (where $F_2^* = F_2$).*

PROOF OF LEMMA 10. Given $\Delta, \epsilon > 0$ we start by applying Lemma 15 to get $\rho = \rho(\epsilon)$ and then apply lemma 14 with that ρ. Thus, we get that for all $\zeta \leq \zeta_0(\Delta, \rho)$, there exist $\delta = \delta_0(\Delta, \rho, \zeta) > 0$, and $\mu = \mu_0(\Delta, \rho, \zeta) > 0$ such that, w.h.p. $\mathcal{F}^n_{\epsilon,\Delta}$ contains no δ-poor, (μ, ζ)-pure subformula. To conclude the proof we set $\zeta_1(\Delta, \epsilon) = \zeta_0(\Delta, \rho(\epsilon))$ and similarly for δ_1, μ_1. □

5. PROOF OF LEMMA 14

Let $\rho, \Delta > 0$, F_2^* be fixed and choose F^* as in the statement of the lemma. For any $\delta, \zeta, \mu > 0$, consider any (candidate) δ-poor, (μ, ζ)-pure subformula H of F^*. Let $v = |V(H)|$ and denote by H_2 the subformula induced by the 2-clauses of H.

The general idea of the argument is as follows. The subformula H_2 has many loose ends, the pure items of H_2, that must be (mostly) covered by the 3-clauses of H in order to for H to have very few pure items. More generally, H may itself create new loose ends but then must cover most of them up again. We show we can cover the literals appearing in H by the clans of the loose ends. Further since the clan sizes are typically small we get a large number of loose ends, the set $P(H)$ below, each of which must be covered by a different 3-clause literal.

In the case that the set of variables in H is large, we will set parameters so that the contribution of all large clans is so small that the δ-poverty of H simply doesn't allow enough 3-clause literals to cover enough loose ends. In the case that the set of variables in H is not so large, we get to use the fact that *every* loose end must be covered but we need a probability argument. Intuitively, it seems unlikely that the random 3-clauses of the formula will exactly cover all loose ends, both those from H_2 and the ones that the 3-clauses themselves generate. One subtlety of the probability analysis is that the new loose ends generated depend on the 3-clauses themselves. In order to make this analysis work, we need stronger bounds on the set of variables in H; we show that, except possibly for the case of the very small number of clans that are not tree-like, we get two loose ends per clan instead of just one and this is enough to make the probability calculation work. Finally, we use a sharp specialized moment bound to show that the rare large clans do not skew the probabilities too much and derive the claimed result.

We now work through the details of the argument. Define the set $P = P(H)$ of literals based on H as follows: P consists of the pure literals of H_2, the smallest numbered literal in each pure cycle of H_2, and every literal on the variables of $V(H) - V(H_2)$. Clearly P contains every pure literal of H and also contains one literal from each pure cycle of H (and since pure cycles are disjoint they are represented by distinct literals).

LEMMA 16. *For any subformula H of F^*, the number of distinct literals in the 3-clauses of H is at least the number of literals in $P(H)$ that are not contained in pure items in H.*

PROOF. We define a one-to-one (but not necessarily onto) mapping from the literals of $P = P(H)$ that are not contained in pure items of H to the literals appearing in the 3-clauses of H. Any literal x in P, that was pure in H_2 or is a literal on $V(H) - V(H_2)$ but is not pure in H, must have $\bar{x}$ in some 3-clause of H and so we map x to $\bar{x}$. The pure cycles of H_2, whose smallest numbered literals form the remainder of P, are disjoint from each other and from the other literals in P. Consider such a cycle C that is pure in H_2 and let $x \in P$ be the smallest numbered literal in C. C will remain pure in H unless there is some y in C such that $\bar{y}$ appears in a 3-clause of H. We map x to $\bar{y}$. The fact that our map is one-to-one follows from the disjointness property of the cycles. □

LEMMA 17. *For any subformula H of F^*, for each literal x on the variables in $V(H)$, there is some literal $y \in P(H)$ such that $x \in \mathrm{Clan}^*_{F_2}(y)$.*

PROOF. If the literal x is on $V(H) - V(H_2)$ then $x \in P$ so we can take $y = x$. If x is a literal on $V(H_2)$ and $\bar{x}$ appears in H_2, then in the digraph $\vec{D}(H_2)$ walk forward from x as far as possible, since all clans contain at most one cycle, either we reach a sink of $\vec{D}(H_2)$, in which case the label of that sink is a pure literal y in P which satisfies $x \in \mathrm{In}_{H_2}(y) \subseteq \mathrm{In}_{F_2}(y)$, or we reach a pure cycle of H_2 in which case the smallest numbered literal y in this cycle satisfies $x \in \mathrm{In}_{H_2}(y) \subseteq \mathrm{In}_{F_2^*}(y)$. Clearly in either case, both x and $\bar{x}$ appear in $\mathrm{Clan}_{F_2^*}(y)$. □

For convenience throughout the rest of this proof we will write $\mathrm{Clan}(x)$ for $\mathrm{Clan}_{F_2^*}(x)$ and for a set T of literals we will write $\mathrm{Clan}(T) = \bigcup_{x \in T} \mathrm{Clan}(x)$.

LEMMA 18. *For any $\zeta \le 1$, there are $\mu_0 = \mu_0(\zeta, \rho)$ and $\delta_0 = \delta_0(\zeta, \rho) > 0$ such that if $\mu \le \mu_0$, $\delta \le \delta_0$, and $\zeta n/2 \le v = |V(H)|$ then H is not a μ-pure subformula of F^*.*

PROOF. Suppose that H is a μ-pure subformula of F^* and $\zeta n/2 \le V(H)$. Since H has at most δv 3-clauses and at most μv items remain pure in H, by Lemma 16, $|P| \le (3\delta + \mu)v$. Choose $I = I(\zeta, \rho)$ such that $\sum_{i>I} 2i(1-\rho)^i < \zeta/2$ and let $\mu_0 = \mu_0(\zeta, \rho) = 1/(4I)$ and $\delta_0 = \delta_0(\zeta, \rho) = 1/(4I)$. Therefore $|P| \le v/I$. By Lemma 17, $|\mathrm{Clan}(P)| \ge 2v$. Let $L_I = \{x \ : \ |\mathrm{Clan}(x)| > I\}$. Using the bound on the sizes of $T_i(F_2^*)$, $|\mathrm{Clan}(P \cap L_I)| \le |\mathrm{Clan}(L_I)| \le \sum_{i>I} 2i(1-\rho)^i n < \zeta n/2 \le v$. Therefore $|\mathrm{Clan}(P - L_I)| \ge |\mathrm{Clan}(P) - \mathrm{Clan}(P \cap L_I)| > v$. However, $|\mathrm{Clan}(P - L_I)| \le |P| \cdot I \le v$ which is a contradiction. □

It remains to show that the probability there is a 0-pure subformula H of F^* with $|V(H)| < \zeta n/2$ is $o(1)$ in n. Suppose that H is 0-pure and $v = |V(H)| < \zeta n/2$. For any literal x (set of literals T), let $cover(x)$ (resp. $cover(T)$) be the set of literals appearing in $\mathrm{Clan}(x)$ (resp. $\mathrm{Clan}(T)$) together with their complements.

LEMMA 19. *For any subformula H of F^* that is 0-pure there exists $P^* = P^*(H) \subseteq P = P(H)$ such that*

1. *$cover(P^*)$ contains every literal appearing in H and*
2. *$|P^*| \le \lfloor \frac{1}{2}(|P| + t_c) \rfloor$ where $t_c = t_c(H)$ is the number of literals $x \in P^*$ such that $\mathrm{In}_{H_2}(x)$ is not tree-like.*

PROOF. Let $\hat{P} \subseteq P$ be the set of literals in P on variables in $V(H_2)$. By definition, for every $x \in P - \hat{P}$, $\bar{x} \in P - \hat{P}$. Let $P_{tree} \subseteq \hat{P}$ be the set of all literals $x \in \hat{P}$ with $\mathrm{In}_{F_2^*}(x)$ tree-like. First we prove that for every $x \in P_{tree}$ there is at least one $y \in \hat{P}$, $y \ne x$ such that $\bar{y} \in \mathrm{In}_{F_2^*}(x)$. For $x \in P_{tree}$, $\mathrm{In}_{H_2}(x)$ is tree-like since $\mathrm{In}_{H_2}(x) \subseteq \mathrm{In}_{F_2^*}(x)$. Therefore there is a vertex $z \in \mathrm{In}_{H_2}(x)$ of in-degree 0 in $\vec{D}(H_2)$ such that $z \ne \bar{x}$. Furthermore, since z appears in H_2, $\bar{z} \in \hat{P}$ so we can take $y = \bar{z} \notin \{x, \bar{x}\}$.

Note that $\bar{y} \in \mathrm{In}_{H_2}(x) \subseteq \mathrm{In}_{F_2^*}(x)$ implies $\bar{x} \in \mathrm{In}_{F_2^*}(y)$. Thus we form an undirected graph G with vertex set $\hat{P}$ and an edge $\langle x, y \rangle$ for each pair of literals with $\bar{y} \in \mathrm{In}_{F_2^*}(x)$. Let $P' \supseteq P_{tree}$ be the set of vertices in G of positive degree, consider a spanning forest of the vertices in P', and consider any bipartition of that forest. Let P_1 be the smaller side of that bipartition. Therefore P_1 dominates P', i.e. every vertex in $P' - P_1$ has a neighbor in P_1 and thus $P_1 \cup (\hat{P} - P')$ dominates all of $\hat{P}$. Letting $|\hat{P} - P'| = a$, $|P_1 \cup (\hat{P} - P')| \le a + \lfloor \frac{1}{2}(|\hat{P}| - a) \rfloor \le \lfloor \frac{1}{2}(|\hat{P}| + a) \rfloor$. Adding the positive form of each literal in $P - \hat{P}$ to $P_1 \cup (\hat{P} - P')$ we obtain a set P^* of size at most $\lfloor \frac{1}{2}(|P| + a) \rfloor$. Since $\hat{P} - P' \subseteq P^*$ and $P_{tree} \subseteq P'$, $t_c \ge a$ and P^* satisfies the claimed size condition.

By definition of P, $\hat{P}$, and P^*, P^* contains the positive literal corresponding to each variable in $V(H) - V(H_2)$, so $cover(P^*)$ contains all literals on variables in $V(H) - V(H_2)$. Further for every literal x such that $\bar{x}$ appears in H_2, as shown in the proof of Lemma 17, $x \in \mathrm{In}_{H_2}(y)$ for some literal y in P. By definition of P^* either $y \in P^*$ or there is some $z \in P^*$ such that $\bar{y} \in \mathrm{In}_{H_2}(z)$. Therefore $x \in \mathrm{Clan}(z)$ and thus both x and $\bar{x}$ are in $cover(z)$. Thus $cover(P^*)$ contains all literals on $V(H_2)$ as well, so the lemma follows. □

We will bound the probability that F has a 0-pure subformula H by bounding $\Pr[P^*(H) = T]$ for each set of literals T and summing over all choices of T.

LEMMA 20. *Let T be a set of literals, $t = |T|$, and suppose that H is a 0-pure subformula of F^* with $P^*(H) = T$ and $t_c = t_c(H)$. Then H and thus formula F^* must contain at least $2t/3 - t_c/3$ 3-clauses whose literals are contained in $cover(T)$; further if $t \leq 10t_c$ then there are at least $3t/5$ 3-clauses of F^* whose literals are contained in $cover(T)$.*

PROOF. By Lemma 19, since $P^*(H) = T$, $|P(H)| \geq 2|T| - t_c = 2t - t_c$. By Lemma 16, since H is 0-pure the 3-clauses of H contain at least $|P(H)| = 2t - t_c$ literals and therefore H contains at least $(2t - t_c)/3$ 3-clauses of F^*. By Lemma 19, all literals in these clauses are in $cover(P^*) = cover(T)$. In case $t \geq 10t_c$ then this is at least $(2/3 - 1/30)t > 3t/5$. □

LEMMA 21. *Fix $\Delta, \rho > 0$. There is $K = K(\Delta)$ such that for T a set of literals, $t = |T|$, the probability that a random F^* has a 0-pure subformula H with $P^*(H) = T$ is at most*

1. *$R(T) = (K/(tn^2))^{3t/5} |\mathrm{Clan}(T)|^{9t/5}$ if $t_c(H) \leq t/10$, and at most*
2. *$R'(T, t_c) = (K/n^2)^{2t/3 - t_c/3} |\mathrm{Clan}(T)|^{2t - t_c}$ if $t_c = t_c(H) > t/10$.*

PROOF. Since F^* has Δn 3-clauses, the probability that at least s of them land entirely in $cover(T)$ is at most

$$\begin{aligned} \binom{\Delta n}{s} \left[\frac{|cover(T)|^3}{8\binom{n}{3}} \right]^s &\leq \binom{\Delta n}{s} \left[\frac{|\mathrm{Clan}(T)|^3}{\binom{n}{3}} \right]^s \\ &\leq [K'/(sn^2)]^s |\mathrm{Clan}(T)|^{3s} \end{aligned}$$

for some constant $K' = K'(\Delta)$. Let $K = 5K'/3$. By Lemma 20, if $t_c(H) \leq t/10$ then we get the probability upper bound in part 1 by setting $s = 3t/5$ and if $t_c(H) > t/10$ then we get the probability upper bound in part 2 by setting $s = 2t/3 - t_c/3$ and observing that $s \geq 1$. □

LEMMA 22. *Fix $\Delta, \rho > 0$. The probability that there exists some set T of literals and a 0-pure subformula H of F^* with $P^*(H) = T$, and $|T| \leq 10t_c(H)$ is $o(1)$ in n.*

PROOF. Let $t \leq 10t_c$ and suppose that H is a 0-pure subformula of F^* with $t_c(H) = t_c$ and $t = |P^*(H)|$. By assumption about F_2^*, $t_c \leq \log n$ so $t \leq 10 \log n$ and thus $|\mathrm{Clan}(P^*(H))| \leq K_0 \log^2 n$ for some constant $K_0 = K_0(\rho)$ because our assumption about the sizes of the $T_i(F_2^*)$ implies that there are no clans of size $\omega(\log n)$. For each $t \leq 10 \log n$ and each t_c, $t_c \leq t$, there are at most $\binom{\log n}{t_c}\binom{2n}{t - t_c}$ different sets T with $|T| = t$ containing t_c literals x with non-tree-like $\mathrm{In}_{F_2^*}(x)$. Therefore, by Lemma 21.2, the probability that there is some 0-pure subformula H of F^* with $|P^*(H)| = t$ and $|t_c(H)| = t_c$ is at most

$$(\log n)^{t_c} (2n)^{t - t_c} (K/n^2)^{2t/3 - t_c/3} (K_0 \log^2 n)^{2t - t_c}$$

which is bounded by $(K'' \log n)^{4t} n^{-t/3}$ for some constant $K'' = K''(\Delta, \rho) > 0$. The probability that an H satisfying the conditions of the lemma exists is then at most

$$\sum_{t_c=1}^{\log n} \sum_{t=t_c}^{10t_c} ((K'')^2 n^{-1/3} \log^4 n)^t \leq 10(K'')^2 n^{-1/3} \log^6 n$$

for n sufficiently large, which is $o(1)$ in n. □

It will be convenient to rewrite the summations over all possible choices of set $T = P^*(H)$ with $|T| = t$ in terms of a probability calculation involving a uniformly chosen random set of literals, T, of size t.

LEMMA 23. *Fix $\rho > 0$. There is a constant $B = B(\rho) > 0$ such that for any $t > 0$ and for T a set of literals with $|T| = t$ chosen uniformly at random, $\mathbf{E}_T(|\mathrm{Clan}(T)|) \leq Bt$.*

PROOF. Let $B = \sum_{i \geq 1} i(1 - \rho)^i$. By assumption, for x chosen uniformly at random from among the $2n$ possible literals, $\mathbf{E}_x(|\mathrm{Clan}(x)|) \leq \sum_{i \geq 1} i(1 - \rho)^i = B$ and therefore $\mathbf{E}_T(|\mathrm{Clan}(T)|) \leq |T| \mathbf{E}_x(|\mathrm{Clan}(x)|) \leq Bt$. □

LEMMA 24. *For every $\rho > 0$ there exists $\alpha = \alpha(\rho) > 0$ such that for all $r \geq 0$ we have for T a set of literals with $|T| = t$ chosen uniformly at random,*

$$\mathbf{Pr}_T(|\mathrm{Clan}(T)| > (r + 16)\mathbf{E}_T(|\mathrm{Clan}(T)|)) < 2 \cdot e^{-\alpha\sqrt{r}t}.$$

PROOF. See appendix. □

LEMMA 25. *Fix $\rho > 0$. There is $K_1 = K_1(\rho)$ such that for any $t > 0$ and for a set of literals T with $|T| = t$ chosen uniformly at random, $E_T(|\mathrm{Clan}(T)|^{9t/5}) \leq (K_1 t)^{9t/5}$.*

PROOF. Fix an integer t and consider choosing T uniformly at random with $|T| = t$. We divide up the range of possible values of $|\mathrm{Clan}(T)|$ into segments of size $\nu(T) = \mathbf{E}_T(|\mathrm{Clan}(T)|) \leq Bt$ where $B = B(\rho)$ is the constant from Lemma 23 and use our tail bounds within each segment. Therefore by Lemma 24,

$$\begin{aligned} &\mathbf{E}_T(|\mathrm{Clan}(T)|^{9t/5}) \\ &\quad\leq 16(\mathbf{E}_T(|\mathrm{Clan}(T)|))^{9t/5} \\ &+ \sum_{r \geq 0} \mathbf{Pr}_T(|\mathrm{Clan}(T)| > (r + 16)\nu(T)) \times [(r + 17)\nu(T)]^{9t/5} \\ &\quad\leq [\nu(T)]^{9t/5} \times (16 + 2 \cdot \sum_{r \geq 0} e^{-\alpha\sqrt{r}t} (r + 17)^{9t/5}) \\ &\quad\leq (Bt)^{9t/5} \times \left(\frac{K_1}{B} \right)^{9t/5} \\ &\quad\leq (K_1 t)^{9t/5}, \end{aligned}$$

for some $K_1 = K_1(\alpha, B) = K_1(\rho)$. □

LEMMA 26. *Fix $\Delta, \rho > 0$. There is $\zeta_0 = \zeta_0(\Delta, \rho) > 0$ such that the probability that a random F^* has a 0-pure subformula H with $v = |V(H)| \leq \zeta_0 n/2$ is $o(1)$ in n.*

PROOF. By Lemmas 21.1 and 22, the probability of this event is at most $\sum_{T,|T|=t\geq 1} R(T)$ plus a term that is $o(1)$ in n. By Lemma 25,

$$\begin{aligned}\sum_{T,|T|=t} R(T) &\leq \binom{2n}{t}(K/(tn^2))^{3t/5}E_T(|\mathrm{Clan}(T)|^{9t/5})\\ &\leq (2en/t)^t(K/(tn^2))^{3t/5}(K_1t)^{9t/5}\\ &= ((2e)^5K^3K_1^9t/n)^{t/5}\\ &\leq (K_2t/n)^{t/5}\end{aligned}$$

for some constant $K_2 = K_2(\Delta,\rho) > 0$.

Now if we let $\zeta_0 = 1/(32K_2)$ then for $v \leq \zeta_0 n/2$, $t \leq v \leq n/(32K_2)$. Therefore by Lemma 21, the probability that such an H exists is at most

$$\sum_{t=1}^{n/(32K_2)}(K_2t/n)^{t/5} = \sum_{t=1}^{\log n}(K_2t/n)^{t/5} + \sum_{t=\log n+1}^{n/(32K_2)}(K_2t/n)^{t/5}.$$

The first summation totals less than $(K_2n^{-1}\log^2 n)^{1/5}$ which is $o(1)$ in n and the second summation totals at most $\sum_{t>\log n} 2^{-t} \leq 2/n$ which is $o(1)$ in n. □

Lemma 14 follows immediately from Lemmas 18 and 26.

6. IMPLICATIONS FOR SATISFIABILITY ALGORITHMS

We now analyze natural DPLL algorithms that are the backtracking versions of several card-type algorithms described below. During the execution of any such algorithm a partial assignment may produce clauses of size 1 (unit clauses) which in turn force additional choices in the partial assignment. The choices by the algorithm made when there are no unit clauses are *free* choices. In UC this free choice is a random assignment to a random unassigned variable; in ORDERED-DLL this is an assignment of 0 to the smallest-numbered unassigned variable and in GUC this is the assignment that satisfies a random literal in a random clause of smallest size.

As the algorithm searches, if the first path in the tree search fails and finds a contradiction, a DPLL algorithm backtracks, undoing the forced choices up to the last free choice, flips the assignment to that variable, calls it forced, and then continues. At this point there are many options for how to continue; probably the simplest option would be to act as if the algorithm had reached this point without backtracking and apply the original heuristic. An alternative heuristic we call FS-backtracking (inspired by [18]) is the following: When a contradiction is reached, record the portion of the assignment of the assignment between the last unexplored free choice and the contradiction; these literals become *hot*. After flipping the value of the last unexplored free choice, instead of making the choice that the original heuristic would suggest, give priority to the complements of the hot literals in the order that they appeared; once the hot literals are exhausted continue as with the original heuristic. FS-backtracking is quite natural in that this last part of the partial assignment got us into trouble in the first place.

Given card-type algorithm A, we write A-FS for the DPLL algorithm extending A using FS-backtracking. Initial experiments comparing ORDERED-DLL-FS to the simple backtracking extension of ORDERED-DLL on random formulas at ratios between 3.8 and 4.0 show that the histogram of run-times of FS-backtracking is *significantly better* than that of simple backtracking throughout the range. The main property of FS-backtracking that is useful in our analysis, as in that of [18], is that at any time when the value of any variable in the partial assignment has flipped at most once during the algorithm's execution, the reduced formula is uniformly random conditional on the number of clauses of each size.

We now give the main ideas for the DPLL lower bounds. Define a *stage* during the execution of a DPLL algorithm to be the time during which the partial assignment of values to variables is constant. A *t-stage* is a stage in which the value of precisely t variables has been set. The "residual formula" in a given stage is the formula that results by removing all clauses that are satisfied by the current (partial) assignment and shrinking all other clauses appropriately.

DEFINITION 27. *Let $\epsilon = 10^{-4}$. A t-stage of a DPLL algorithm is* bad *if the residual formula at that stage is distributed as the union of a random 3-CNF formula with $(2.281 \pm \epsilon)t$ clauses and a random 2-CNF formula with $(0.999 \pm \epsilon)t$ 2-clauses.*

LEMMA 28. *Let $r_{\mathrm{UC}} = r_{\mathrm{ORDERED\text{-}DLL}} = 3.81$ and let $r_{\mathrm{GUC}} = 3.98$. For each $A \in \{$UC,ORDERED-DLL,GUC$\}$, there exists $p_A > 0$ such that an execution of algorithm $A \in \{$UC,ORDERED-DLL,GUC$\}$ on a random 3-CNF formula with r_An clauses reaches a* bad *stage with $t \leq n/2$ with probability at least p_A. For each $A \in \{$UC,ORDERED-DLL,GUC$\}$, an execution of algorithm A-*RS *on a random 3-CNF formula with r_An clauses reaches a* bad *stage with $t \leq n/2$ w.h.p.*

COROLLARY 29. *Let $r_{\mathrm{UC}} = r_{\mathrm{ORDERED\text{-}DLL}} = 3.81$ and let $r_{\mathrm{GUC}} = 3.98$. For $A \in \{$UC,ORDERED-DLL,GUC$\}$, there exists $p_A > 0$ such that an execution of* any *backtracking extension of algorithm $A \in \{$UC,ORDERED-DLL,GUC$\}$ on a random 3-CNF formula with r_An clauses takes time $2^{\Omega(n)}$ with probability at least p_A. For $A \in \{$UC,ORDERED-DLL,GUC$\}$, an execution of algorithm A-*RS *on a random 3-CNF formula with r_An clauses takes time $2^{\Omega(n)}$ w.h.p.*

7. FURTHER RESEARCH

Our upper bounds on the number of 3-clauses needed to cause exponential behavior in satisfiability algorithms will be readily improved with any improvement on the $2.28n$ upper bound for unsatisfiability in random $(2+p)$-SAT. That is, if it is shown that for some $\epsilon > 0$ and $2/3 \leq r < 2.28$, random formulas with $(1-\epsilon)n$ 2-clauses and rn 3-clauses are unsatisfiable w.h.p. then the bounds of 3.81 and 3.98 will be immediately reduced. In fact, if r is reduced to $2/3$, to match the lower bound, then our results immediately imply the following remarkably sharp behavior: every card-type algorithm A is such that it operates in linear time with constant probability up to some threshold β_A but any backtracking extension of A requires exponential time with constant probability for all ratios larger than β_A. In fact, if A uses FS-backtracking then it would work in linear time almost surely at ratios below β_A and require exponential time almost surely above β_A.

It seems quite likely that one can extend our w.h.p. analysis to the simple backtracking versions of UC, GUC, ORDERED-DLL, and other card-type algorithms.

8. REFERENCES

[1] D. Achlioptas. A survey of lower bounds for random 3-SAT via differential equations. *Theoret. Comput. Sci., to appear*. Available as
http://www.research.microsoft.com/∼optas/lbsurvey.ps.

[2] D. Achlioptas. *Threshold Phenomena in random graph coloring and satisfiability*. PhD thesis, Department of Computer Science, University of Toronto, 1999.

[3] D. Achlioptas, A. Chtcherba, I. Istrate, and C. Moore. The phase transition in 1-in-k SAT and NAE 3-SAT. In *12th Annual ACM-SIAM Symposium on Discrete Algorithms (Washington, DC, 2001)*, pages 721–722. ACM, New York, 2001.

[4] D. Achlioptas, L. M. Kirousis, E. Kranakis, and D. Krizanc. Rigorous results for random $(2+p)$-SAT. *Theoret. Comput. Sci., to appear.*

[5] D. Achlioptas and G. B. Sorkin. Optimal myopic algorithms for random 3-SAT. In *41st Annual Symposium on Foundations of Computer Science (Redondo Beach, CA, 2000)*, pages 590–600. IEEE, 2000.

[6] P. Beame, R. Karp, T. Pitassi, and M. Saks. On the complexity of unsatisfiability of random k-CNF formulas. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pages 561–571, Dallas, TX, May 1998.

[7] E. Ben-Sasson and R. Impagliazzo. Random CNF's are hard for the polynomial calculus. In *Proceedings 40th Annual Symposium on Foundations of Computer Science*, pages 415–421, New York,NY, October 1999. IEEE.

[8] E. Ben-Sasson and A. Wigderson. Short proofs are narrow – resolution made simple. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pages 517–526, Atlanta, GA, May 1999.

[9] A. Biere, A. Cimatti, E. Clarke, and Y. Zhu. Symbolic model checking without BDDs. In *Procceedings, 5th International Conference, TACAS'99*, pages 193–207, Berlin, Germany, 1999. Springer-Verlag.

[10] B. Bollobás. *Random graphs*. Academic Press, London-New York, 1985.

[11] M.T. Chao and J. Franco. Probabilistic analysis of a generalization of the unit-clause literal selection heuristics. *Information Science*, 51:289–314, 1990.

[12] 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.

[13] V. Chvátal and E. Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, 1988.

[14] C. Coarfa, D. D. Demopoulos, A. San Miguel Aguirre, D. Subramanian, and M. Y. Vardi. Random 3-SAT: The plot thickens. In *Proceedings 6th International Conference on Principles and Practice of Constraint Programming*, Singapore, September 2000.

[15] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem proving. *Communications of the ACM*, 5:394–397, 1962.

[16] M. Davis and H. Putnam. A computing procedure for quantification theory. *Communications of the ACM*, 7:201–215, 1960.

[17] E. Friedgut. Sharp thresholds of graph properties, and the k-sat problem. *Journal of the American Mathematical Society*, 12:1017–1054, 1999.

[18] A. Frieze and S. Suen. Analysis of two simple heuristics on a random instance of k- SAT. *Journal of Algorithms*, 20(2):312–355, 1996.

[19] A. Goerdt. A threshold for unsatisfiability. *Journal of Computer and System Sciences*, 53:469–486, 1996.

[20] C. Gomes, B. Selman, and H. Kautz. Boosting combinatorial search through randomization. In *Proceedings Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 431–437, 1998.

[21] C. P. Gomes, B. Selman, N. Crato, and H. Kautz. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *J. Automat. Reason.*, 24(1-2):67–100, 2000.

[22] S. Janson, Y. C. Stamatiou, and M. Vamvakari. Bounding the unsatisfiability threshold of random 3-SAT. *Random Structures Algorithms*, 17(2):103–116, 2000.

[23] H. Kautz, D. McAllester, and B. Selman. Encoding plans in propositional logic. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference (KR'96)*, pages 374–384, 1996.

[24] H. Kautz and B. Selman. Pushing the envelope: planning, propositional logic, and stochastic search. In *Proceedings of the 13th AAAI*, pages 1194–2001, 1996.

[25] S. Kirkpatrick and B. Selman. Critical behavior in the satisfiability of random formulas. *Science*, 264:1297–1301, May 1994.

[26] C. J.H. McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics, Proceedings of the 12th British Combinatorial Conference*, pages 148–188. 1989.

[27] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky. Phase transition and search cost in the $(2+p)$-SAT problem. In *4th Workshop on Physics and Computation, (Boston, MA, 1996)*.

[28] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky. $2+p$-SAT: relation of typical-case complexity to the nature of the phase transition. *Random Structures Algorithms*, 15(3-4):414–435, 1999.

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

[30] B. Selman and H. Kautz. Domain-independent extensions to GSAT: Solving large structured satisfiability problems. In *Proceedings of the 13th IJCAI*, pages 290–295, 1993.

[31] B. Selman, H. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. In *Proceedings Tenth National Conference on Artificial Intelligence (AAAI-92)*, pages 440–446, 1992.

[32] B. Selman, D. Mitchell, and H. Levesque. Generating hard satisfiability problems. *Artificial Intelligence*, 81:17–29, 1996.

APPENDIX

A. PROPERTIES OF SUBCRITICAL RANDOM 2-CNF FORMULAE

We will now prove that subcritical random 2-CNF formulas satisfy the properties in Lemma 14 w.h.p.

LEMMA 30. *Let F_2 be random 2-SAT formula formed by picking $m_2 = (1-\epsilon)n$ clauses from $C_2(n)$ uniformly, independently and with replacement. There exists $\rho = \rho(\epsilon) > 0$ such that w.h.p. all of the following hold simultaneously.*

1. *For every literal ℓ, $\mathrm{In}_{F_2}(\ell)$ is simple.*
2. *There at most $\log n$ literals ℓ, such that $\mathrm{In}_{F_2}(\ell)$ is not tree-like.*
3. *For all i, $T_i(F_2) \leq 2(1-\rho)^i\, n$.*

PROOF. (*Sketch*) We will choose ρ later as it only affects proving 3. Let $p = (1-\epsilon)/n$ and consider the random graph $G(n,p)$ on n vertices, where each of the $\binom{n}{2}$ possible edges appears independently with probability p. Fix a vertex v in $G(n,p)$ and consider the random subgraph $C(v)$ that corresponds to the connected component of v. Now, fix a literal ℓ in F_2 and consider the undirected graph $G(\mathrm{In}_{F_2}(\ell))$. It is not hard to show that one can couple the random graphs $C(v)$ and $G(\mathrm{In}_{F_2}(\ell))$ so that $G(\mathrm{In}_{F_2}(\ell)) \subseteq C(v)$ always, i.e. $G(\mathrm{In}_{F_2}(\ell))$ is always a subgraph of $C(v)$. This coupling allows us to exploit a number of well-known facts about the components of subcritical random graphs, i.e. $G(n, p = (1-\epsilon)/n)$.

Proof of 1: Let R denote the set of all literals ℓ such that $G(\mathrm{In}_{F_2}(\ell))$ contains more than one cycle. It is well-known [10] that $\Pr[C(v)$ contains more than one cycle$] < B/n^{3/2}$ for some $B = B(\epsilon)$. Thus, $\mathbf{E}(|R|) = o(1)$ and hence w.h.p. $R = \emptyset$.

Proof of 2: Let Q denote the set of all literals ℓ such that $G(\mathrm{In}_{F_2}(\ell))$ contains at least one cycle. It is well-known [10] that $\Pr[C(v)$ contains at least one cycle$] < C/n$ for some $C = C(\epsilon)$.Thus, $\mathbf{E}(|Q|) < 2C$ and the claim follows from Markov's inequality.

Proof of 3: We will first establish that there exists $\sigma = \sigma(\epsilon)$ such that for any fixed literal ℓ and all $q \geq 1$,

$$\Pr[|\mathrm{Clan}_{F_2}(\ell)| = q] < (1-\sigma)^q \ .$$

For that we observe that for a fixed literal ℓ one can construct a random variable W that dominates $|\mathrm{Clan}_{F_2}(\ell)|$ as follows. Let $X \overset{D}{=} |C(v)|$, where v is a fixed vertex in $G(n, p = (1-\epsilon)/n)$ and $|C(v)|$ denotes the number of vertices in its connected component $C(v)$. Now, let

$$W = S_1 + S_2 + \cdots + S_X$$

where the S_i are i.i.d. random variables with $S_i \overset{D}{=} |C(v)|$. To bound the upper tail of W we will use a standard moment generating function argument.

Let $\Pr[S_i = k] = \Pr[X = k] \equiv p_k$ and let $c = 1 - \epsilon$. Using that, asymptotically in n,

$$p_k = (1/c)\left(ce^{-c}\right)^k k^{k-1}/k! \tag{3}$$

and that for all $c < 1$,

$$\sum_{k\geq 1} p_k = 1 \ , \tag{4}$$

with a bit of work it follows that for any $h > 0$,

$$\begin{aligned} \mathbf{E}(\exp(hW)) &< \sum_{k\geq 1} p_k (d/c)^k \\ &= (1/c)\sum_{k\geq 1}(de^{-c})^k k^{k-1}/k! \ < \ 1/c \ . \end{aligned}$$

From this it's rather straightforward to show that for all $\epsilon > 0$ there is $\tau = \tau(\epsilon) > 0$ and $q_0 = q_0(\epsilon)$ such that for all $q \geq q_0$,

$$\begin{aligned} \Pr[|\mathrm{Clan}_{F_2}(\ell)| = q] &\leq \Pr[W = q] \\ &< \frac{1}{1-\epsilon}\exp\left(-\frac{\epsilon^3}{3}q\right) \leq (1-\tau)^q \ . \end{aligned}$$

Since for any fixed value of q, $\Pr[|\mathrm{Clan}_{F_2}(\ell)| = q] < 1$, it follows that there exists $\sigma = \sigma(\tau, q_0)$ such that for all $q \geq 1$,

$$\Pr[|\mathrm{Clan}_{F_2}(\ell)| = q] < (1-\sigma)^q \ . \tag{5}$$

By linearity of expectation and (5) we get that for all i,

$$\mathbf{E}\left(T_i(F_2)\right) < 2n \times (1-\sigma)^i \ . \tag{6}$$

Our next step is to show that for each i there exists $Q_i \leq \mathbf{E}\left(T_i(F_2)\right)$ such that w.h.p. for all i,

$$|T_i(F_2) - Q_i| < n^{3/4} \ . \tag{7}$$

Let us first observe that (5) also implies that there exists $C = C(\epsilon)$ such that w.h.p.

$$\text{for all } \ell \text{ in } F_2, \quad |\mathrm{Clan}_{F_2}(\ell)| < C\log n \ . \tag{8}$$

To prove (7) we will need to do some work before appealing to a concentration inequality. The reason for this is that, a priori, replacing a single clause in F_2 could change $T_i(F_2)$ dramatically, for some i; luckily, this is an unlikely event. To capture this last fact we will introduce a family of random variables U_i with the following properties: i) w.h.p. $U_i(F_2) = T_i(F_2)$ for all i, and ii) by definition (of the U_i), replacing a clause in F_2 can affect each U_i by at most $\mathrm{polylog}(n)$. We omit the construction of the random variables U_i from this extended abstract. The rough idea is that they correspond to the T_i if we were number the appearances of clauses in clans and ignore all but the first $\log n$ appearances of each clause. By construction, the U_i do not suffer from the possibility of a single clause making a dramatic difference in their value and hence concentration follows from standard martingale arguments.

Combinining (6) and (7) we get that there exists $\sigma = \sigma(\epsilon) > 0$ such that w.h.p.

$$T_i \leq 2n \times (1-\sigma)^i + n^{3/4} \ . \tag{9}$$

Further, recall that by (5)

$$\text{w.h.p. } T_i = 0 \text{ for all } i \geq C\log n \ . \tag{10}$$

Let us now choose $\phi < \sigma$ such that $(1-\phi)^{C\log n} \geq n^{-1/4}$. Thus, for all $i < C\log n$, $2n \times (1-\phi)^i \geq 2n^{3/4}$. We claim that w.h.p. for all i,

$$T_i \leq (4n) \times (1-\phi)^i \ . \tag{11}$$

If $i \geq C\log n$ then (11) holds by (10). If $i < C\log n$ then by (9), (11) and $\phi < \sigma$, respectively,

$$\begin{aligned} T_i &\leq 2n \times (1-\sigma)^i + n^{3/4} \\ &\leq 2n \times (1-\sigma)^i + 2n \times (1-\phi)^i \\ &\leq 4n \times (1-\phi)^i \ . \end{aligned}$$

By (9) and (11) it follows that there is $\rho < \phi$ such that w.h.p. for all i, $T_i \leq 2n \times (1-\rho)^i$. □

B. PROOF OF LEMMA 24

We will prove a somewhat more general concentration statement, cast in terms of picking weighted balls without replacement. In particular, we assume that we have a set $\mathcal{B}$ of $2n$ weighted balls, each ball x having $\mathrm{weight}(x) \geq 1$. Let T_i denote the number of balls with weight i. Our lemma holds for any weight sequence $T_1, T_2, \ldots$ which satisfies the following condition: there exists $\rho > 0$ such that

$$T_i \leq 2(1-\rho)^i n, \quad \text{for all } i \ . \tag{12}$$

We will pick a random subset $R \subseteq \mathcal{B}$ of $t \leq n$ balls, i.e. we pick randomly without replacement, and let

$$W = \sum_{x\in R} \mathrm{weight}(x) \ .$$

We will prove that there is $\alpha > 0$ such that for every $\xi, t \geq 1$,

$$\Pr[W > 4(1+\xi)^2\mathbf{E}(W)] < 2\exp(-3\alpha\xi t) \ . \tag{13}$$

Lemma 24 will follow from (13) by setting $4(1+\xi)^2 = r+16$ and observing that $\xi = \sqrt{4+r/4} - 1 \geq \max\{1, \sqrt{r}/3\}$.

PROOF. We start by considering W to be defined in the following, equivalent, manner. Let S be an infinite sequence of balls formed by choosing balls uniformly, independently and *with replacement* from $\mathcal{B}$. Let W be the sum of the weights of the first t distinct elements of S.

Let us consider the prefix $P = p_1, p_2, \ldots, p_d$ of S where $d = 2(1+\xi) \times t$. In particular, let us form a random set $R' \subseteq \mathcal{B}$, by scanning P linearly and adding to R' every ball not seen before, until either $|R'| = t$ or we exhaust P. Let

$$W' = \sum_{x \in R'} \text{weight}(x) \qquad \text{and} \qquad Q = \sum_{i=1}^{d} \text{weight}(p_i) \ .$$

Then, by (the miracle of) linearity of expectation, we see that $\mathbf{E}(Q) = 2(1+\xi)\mathbf{E}(W)$ and, thus, for any $\xi > 0$

$$\begin{aligned}
&\Pr[W > 4(1+\xi)^2\mathbf{E}(W)] \\
&\quad\leq \ \Pr[W' > 4(1+\xi)^2\mathbf{E}(W)] + \Pr[W' \neq W] \\
&\quad\leq \ \Pr[Q > 4(1+\xi)^2\mathbf{E}(W)] + \Pr[W' \neq W] \\
&\quad\leq \ \Pr[Q > (2+\xi)\mathbf{E}(Q)] + \Pr[W' \neq W] \ .
\end{aligned}$$

For $W' \neq W$ to occur it must be that we picked $2(1+\xi)t$ balls out of $2n$ balls with replacement and got fewer than t distinct balls. Using standard results for the coupons collector problem, it is not hard to show that since $t \leq n$, there exists $\phi > 0$ such that the probability of this event is $\exp(-\phi\xi^2 t) \leq \exp(-\phi\xi t)$ for $\xi \geq 1$.

We will prove below that $\Pr[Q > (2+\xi)\mathbf{E}(Q)] < \exp(-\theta\xi t)$ for some $\theta = \theta(\rho) > 0$. Combining this with the estimate for $W' \neq W$ we get that for $\xi \geq 1$ the probability of having $W > 4(1+\xi)^2\mathbf{E}(W)$ is at most $\exp(-\theta\xi t) \leq \exp(-\phi\xi t) \leq 2\exp(-3\alpha\xi t)$ for $\alpha = \min\{\phi, \theta\}/3$ as required.

To prove our tail bound on Q we first note that for any $h > 0$,

$$\begin{aligned}
&\Pr[Q > (2+\xi)\mathbf{E}(Q)] \\
&\quad= \ \Pr[\exp(hQ) > \exp((2+\xi)h\mathbf{E}(Q))] \\
&\quad\leq \ \mathbf{E}(\exp(hQ)) \times \exp(-(2+\xi)h\mathbf{E}(Q)) \ . \qquad (14)
\end{aligned}$$

Now let $\{Q_i\}_{i=1}^d$ be i.i.d.r.v. defined by $Q_i = \text{weight}(p_i)$. Thus, $Q = \sum_{i=1}^d Q_i$ and as a result

$$\begin{aligned}
\mathbf{E}(\exp(hQ)) &= \mathbf{E}\left(\prod_{i=1}^{d} \exp(hQ_i)\right) \\
&= \prod_{i=1}^{d} \mathbf{E}\left(\exp\left(hQ_i\right)\right) \qquad (15) \\
&= \left(\mathbf{E}\left(\exp\left(hQ_i\right)\right)\right)^d \ . \qquad (16)
\end{aligned}$$

To simplify notation let us replace Q_i with T in the rest of the proof and let $\mu = \mathbf{E}(T)$.

To go from (17) to (18) we use (12). To go from (18) to (19) we require $h < \rho$, which suffices to guarantee the sum's convergence. Finally, to go from (19) to (20) we use that for $h > 0$, $e^{-h} > 1 - h$.

$$\begin{aligned}
&\mathbf{E}(\exp(hT)) \\
&\quad= \sum_{i=1}^{\infty} \Pr[T = i]\exp(hi) \\
&\quad= \sum_{i=1}^{\infty} \Pr[T = i]\left(1 + hi + (\exp(hi) - hi - 1)\right) \qquad (17) \\
&\quad\leq 1 + h\mu + \sum_{i=1}^{\infty} (1-\rho)^i(\exp(hi) - hi - 1) \qquad (18) \\
&\quad= 1 + h\mu + (1-\rho)\left(\frac{1}{\rho - 1 + \exp(-h)} - \frac{h+\rho}{\rho^2}\right) \qquad (19) \\
&\quad< 1 + h\mu + (1-\rho)\left(\frac{1}{\rho - h} - \frac{h+\rho}{\rho^2}\right) \qquad (20) \\
&\quad= 1 + h\mu + \frac{h^2(1-\rho)}{\rho^2(\rho - h)} \ . \qquad (21)
\end{aligned}$$

Now, substituting $h = \rho^3$ in (21) we get (22), while (23) follows from $\mu \geq 1 > (\rho+1)^{-1}$.

$$\begin{aligned}
\mathbf{E}(\exp(\rho^3 T)) &< 1 + \rho^3\mu + \frac{\rho^3}{\rho+1} \qquad (22) \\
&< 1 + 2\rho^3\mu \ . \qquad (23)
\end{aligned}$$

Note now that, by (14) and (16), for all $h > 0$,

$$\begin{aligned}
&\Pr[Q > (2+\xi)\mathbf{E}(Q)] \\
&\quad\leq \left(\frac{\mathbf{E}(\exp(hT))}{\exp((2+\xi)h\mathbf{E}(T))}\right)^{2(1+\xi)t} \\
&\quad\leq \left(\frac{\mathbf{E}(\exp(hT))}{\exp(2h\mathbf{E}(T))}\right)^{2(1+\xi)t} \times \exp(-2h\xi\mu t) \ . \qquad (24)
\end{aligned}$$

Taking $h = \rho^3$, (23) implies that the ratio in (24) is bounded by 1. Thus, since $\mu \geq 1$, if $\theta = 2\rho^3$, then

$$\Pr[Q > (2+\xi)\mathbf{E}(Q)] \leq \exp(-\theta\xi t) \ .$$

□