SIAM J. COMPUT.
Vol. 39, No. 1, pp. 260–280

RANDOM FORMULAS HAVE FROZEN VARIABLES*

DIMITRIS ACHLIOPTAS† AND FEDERICO RICCI-TERSENGHI‡

Abstract. For a large number of random constraint satisfaction problems, such as random k-SAT and random graph and hypergraph coloring, there exist very good estimates of the largest constraint density for which solutions exist. All known polynomial-time algorithms for these problems, though, fail to find solutions even at much lower densities. To understand the origin of this gap one can study how the structure of the space of solutions evolves in such problems as constraints are added. In particular, it is known that much before solutions disappear, they organize into an exponential number of clusters, each of which is relatively small and far apart from all other clusters. Here we further prove that inside every cluster the vast majority of variables are frozen, i.e., take only one value. The existence of such frozen variables gives a satisfying intuitive explanation for the failure of the polynomial-time algorithms analyzed so far. At the same time, our results lend support to one of the two main hypotheses underlying Survey Propagation, a heuristic introduced by physicists in recent years that appears to perform extraordinarily well on random constraint satisfaction problems.

Key words. satisfiability, random formulas, phase transitions, frozen variables

AMS subject classifications. 68R99, 82B26, 05C80

DOI. 10.1137/070680382

1. Introduction.

For a number of random Constraint Satisfaction Problems (CSP), there exist very good estimates of the largest constraint density (ratio of constraints to variables) for which typical instances have solutions. For example [3], we know that a random graph of average degree d is k-colorable with high probability[1] if $d < (2k-2)\ln(k-1)$ but w.h.p. non-k-colorable if $d > (2k-1)\ln k$. This implies that for every $d > 0$, w.h.p. the chromatic number of a random graph with average degree d is either k_d or $k_d + 1$, where k_d is the smallest integer k such that $d < 2k\ln k$. The proof in [3] is via the second moment method, applied to the number of k-colorings. As such, while it establishes the existence of (exponentially many) k-colorings, it gives no information whatsoever on how to efficiently find one.

Algorithmically, it is very easy to get a factor-2 approximation for the graph coloring problem on random graphs. The algorithm "repeatedly pick a random vertex and assign it a random available color" will w.h.p. succeed in coloring a random graph of average degree d if originally each vertex has $2k_d$ available colors. Alternatively, k colors suffice when $d < k\ln k$. In spite of significant efforts over the last 30 years, no improvement has been made over this trivial algorithm. Specifically, no polynomial-time algorithm is known that k-colors random graphs of average degree $d = (1+\epsilon)k\ln k$, for some fixed $\epsilon > 0$ and arbitrarily large k.

In the random k-SAT problem one asks if a random k-CNF formula, $F_k(n,m)$, with n variables and m clauses is satisfiable. It is widely believed that the probability that such a formula is satisfiable exhibits a sharp threshold. Specifically, the

*Received by the editors January 18, 2007; accepted for publication (in revised form) June 11, 2008; published electronically May 28, 2009.
http://www.siam.org/journals/sicomp/39-1/68038.html
†Department of Computer Science, University of California Santa Cruz, Santa Cruz, CA 95064 (optas@cs.ucsc.edu).
‡Physics Department, University of Rome "La Sapienza," I-00185 Rome, Italy (federico.ricci@roma1.infn.it).
[1]We say that a sequence of events $\mathcal{E}_n$ occurs with high probability (w.h.p.) if $\lim_{n\to\infty}\Pr[\mathcal{E}_n] = 1$.

TABLE 1

k	3	4	7	10	20
Best known upper bound for r_k^*	4.508	10.23	87.88	708.94	726,817
Best known lower bound for r_k	3.52	7.91	84.82	704.94	726,809
Best known algorithmic lower bound	3.52	5.54	33.23	172.65	95,263

Satisfiability Threshold Conjecture asserts that $r_k = r_k^*$ for all $k \geq 3$, where

$$r_k \equiv \sup\{r : F_k(n, rn) \text{ is satisfiable w.h.p.}\},$$
$$r_k^* \equiv \inf\{r : F_k(n, rn) \text{ is unsatisfiable w.h.p.}\}.$$

It is easy to see that $r_k^* \leq 2^k \ln 2$, since the probability that at least one assignment satisfies $F_k(n, rn)$ is bounded by $2^n(1 - 2^{-k})^{rn}$, a quantity that tends to 0 for $r \geq 2^k \ln 2$. In [7] it was shown that random k-CNF formulas w.h.p. have satisfying assignments for densities very close to this upper bound. Specifically, for all $k \geq 3$,

$$r_k > 2^k \ln 2 - \frac{(k+1)\ln 2 + 3}{2}. \tag{1}$$

This lower bound was also derived via the second moment method, similarly to the k-coloring problem. Here, though, the gap relative to algorithms is even greater: no polynomial algorithm is known that finds satisfying assignments in a random k-CNF formula when $r = \omega(k)\, 2^k/k$ for any function $\omega(k) \to \infty$ (arbitrarily slowly). In Table 1, we illustrate this gap for some small values of k. For $k = 3$, the upper bound on r_k^* comes from [14], while for $k > 3$ it is from [13, 20]. The best algorithmic lower bound for $k = 3$ is from [16, 19], while for $k > 3$ it is from [15].

Similar results (and gaps) exist for a number of other CSP, such as random NAE k-SAT and hypergraph 2-coloring, regular random graph coloring, random Max k-SAT, and others (see [2, 4, 5, 6]). Indeed, this phenomenon seems to occur in nearly all random CSP in which the underlying constraint graph is sparse and random, making it natural to ask if there is a common underlying cause. (The bipartite graphs where constraints are adjacent to the variables they bind are also known as factor graphs.)

As it turns out, sparse random CSP have been systematically studied by physicists in the past few decades under the name "mean-field diluted spin-glasses." "Spins" here are the variables (reflecting the notion that variables have small, discrete domains), while the term "glass" refers to the fact that the system has not been allowed to relax to a configuration in which spins interact in a mutually agreeable way (reflecting that different constraints prefer different values for the variables). "Diluted" refers to the fact that the factor graph is sparse (reflecting that each spin interacts with only a few other spins), while "mean field" refers to the fact that the factor graph is random, i.e., there is no underlying geometry mandating the interactions. The physicists' interest in mean-field systems is partly motivated by the fact that for many statistical mechanics problems in which the variables do lie on a lattice, such as $\mathbb{Z}^d$, the effect of the underlying geometry vanishes for d sufficiently large (but finite).

In the last few years, motivated by ideas developed for the study of materials, physicists have put forward a hypothesis for the origin of the aforementioned algorithmic gap in random CSP and, most remarkably, a method for overcoming it. Specifically, Mézard, Parisi, and Zecchina [23] developed an extremely efficient algorithm, called Survey Propagation (SP), for finding satisfying assignments of random

formulas in the satisfiable regime. For example, their algorithm typically finds a satisfying truth assignment of a random 3-CNF formula with $n = 10^6$ variables and $4.25n$ clauses in minutes (and appears to scale as $O(n \log n)$). No other algorithm practically solves formulas of such density with $n = 10^4$.

Our original motivation for this work was to see if some of the physically motivated ideas underlying SP can be proven mathematically. More generally, we believe that understanding the geometry of the space of satisfying truth assignments in random formulas is essential for understanding the behavior of algorithms on them. This is particularly true for random-walk-type algorithms, for which very little is known rigorously, with the notable exception of [9]. We view such algorithms as the first natural class to target armed with an understanding of solution-space geometry.

In [1] we made progress towards this goal by proving that already much below the satisfiability threshold, the set of satisfying assignments fragments into exponentially many connected components (clusters) that are relatively small in size and far apart from one another. Here, we additionally prove that inside each such connected component, the vast majority of variables are "frozen," i.e., take only one value. As the formula density is increased towards the threshold, the fraction of frozen variables in each component increases, causing the connected components to decrease in volume and grow further apart from one another. As we discuss below, the existence of frozen variables provides a good explanation for the origin of the barrier faced by all analyzed algorithms on random CSP, i.e., "local" algorithms.

Our results regarding frozen variables are in perfect agreement with the physics picture and strongly suggest that one of the two main assumptions underlying SP regarding the structure of the set of solutions is essentially correct. This brings us closer to a rigorous analysis of SP and answers affirmatively the main open question raised by Maneva, Mossel, and Wainwright in [22]. Specifically, we prove that for all $k \geq 9$, the connected components of the set of satisfying assignments of random formulas have nontrivial cores, as assumed by SP (see Definition 8). We point out that it is not clear whether this is true for small k. Indeed, [22] gave experimental evidence that for $k = 3$, random formulas do *not* have nontrivial cores. As we will see, our methods also give some evidence in that direction. This gives additional motivation for the "core-like" objects introduced in [22] whose existence would relate to the success of SP for small k (we discuss this point further in section 3.1).

In the next section we give an informal discussion relating the performance of DPLL-type algorithms on random formulas to notions such as Gibbs sampling and long-range correlations. This is meant to provide intuition for the empirical success of SP and motivate our results. We emphasize that while both the discussion and the results are about random k-SAT, this is not strictly necessary: our ideas and proofs are quite generic, and should generalize readily to many other random CSP, e.g., graph coloring.

1.1. DPLL algorithms, belief propagation, and frozen variables. Given a satisfiable formula F on variables $v_1, v_2, \ldots, v_n$, it is easy to see that the following simple procedure samples uniformly from the set of all satisfying assignments of F:

Start with the input formula F.
For $i = 1$ to n do:
1. Compute the fraction, p_i, of satisfying assignments of the current formula in which v_i takes the value 1.

2. Set v_i to 1 with probability p_i and to 0 otherwise.
3. Simplify the formula.

Clearly, the first step in the loop above is meant only as a thought experiment. Nevertheless, it is worth making the following two observations. The first is that if we are only interested in finding *some* satisfying assignment, as opposed to sampling a uniformly random one, then we do not need to compute exact marginals. For example, if we use the rule of setting v_i to 1 iff $p_i \geq 1/2$, then it is enough to ensure that if a variable takes the same value x in *all* satisfying assignments, x is the majority value in our computed marginal. The second observation is that the order in which we set the variables does not need to be determined a priori. That is, we can imagine that in each step we compute marginals for all remaining variables and that for each marginal we have an associated confidence. To improve our chances of avoiding a fatal error, we can then set only the variable for which we have highest confidence.

The above two elementary observations in fact capture all algorithms that have been analyzed so far on random formulas (and, in fact, most DPLL-type algorithms used in practice). Observe, for example, that both the *unit-clause* and the *pure literal* heuristics follow immediately from the above considerations. In the case of unit-clause, the participation of a variable v in a unit-clause c allows us to infer its marginal with perfect confidence and thus setting v is an "obvious" choice. In the case of a pure literal ℓ, we can infer with certainty the majority marginal of the underlying variable v (it is the value that satisfies ℓ). In the absence of such obvious choices, all DPLL-type algorithms attempt to guess the majority marginal of some variable v and set v to that value. For example, below are the choices made in the absence of unit-clauses and pure literals by some of the algorithms that have been analyzed on random 3-CNF formulas. In order of improving performance:

UNIT-CLAUSE [12]: select a random variable and assign it a random value.
3-CLAUSE MAJORITY [11]: select a random variable and assign it its majority value among the 3-clauses.
SHORT-CLAUSE [15]: select a random shortest clause c, a random variable v in c, and set v so as to satisfy c.
HAPPIEST LITERAL [18]: satisfy a literal that appears in most clauses.

Each of the above heuristics attempts to compute marginals based on a different set of evidence, the content of which ranges from completely empty [12] to considering all the clauses containing each variable [18]. Correspondingly, the largest density for which these algorithms succeed on random 3-CNF formulas ranges from 8/3 for [12] to 3.42 for [18]. UNIT-CLAUSE, in fact, succeeds for every k as long as $r < 2^k/k$ and, as we mentioned earlier, no algorithm is known to beat this bound asymptotically. Given that improving upon the empty set of evidence is rather easy, it is tempting to think that by considering a larger set of evidence for each variable one can do significantly better. For example, consider an algorithm $\mathcal{A}_d$ which computes a marginal for each variable v based on the clauses that appear in the depth-d neighborhood of v in the factor graph. One could hope that as d grows, such an algorithm would do well, perhaps even reach the satisfiability threshold.

Physicists predict that this is not the case. The hope that local algorithms could do well on random formulas rests on the presumption that the influence exerted on a variable v by other variables diminishes rapidly with their distance from v in the factor graph. In other words, the hope is that there are no "long range correlations" in random formulas and, as a result, the joint probability distribution of a random finite subset of the variables should be, essentially, the product of their marginals.

Unfortunately, the existence of clusters with numerous frozen variables can induce long-range correlations among the variables, eliminating such hopes. For example, it is predicted in [21] that for every $k \geq 3$, there exists a constant $d_k < r_k$ such that for all densities between d_k and r_k, for every integer $q \geq 1$, with probability $f(q) > 0$, all but a vanishing fraction of satisfying assignments lie in the q (largest) clusters. Since for $k \geq 8$, at these densities all clusters are far apart from one another and have a large fraction of frozen variables (as we will prove here), it follows that the joint density of a fixed set of variables over the uniform measure of solutions fails to factorize. To overcome the above issue, physicists model each connected component of satisfying assignments as a subcube that results by selecting a large fraction of the variables and freezing them independently at random, while leaving the rest (largely) free. Our results imply that this simplified view of clusters is not very far from the truth.

2. Statement of results. We first introduce some definitions. Throughout, we assume that we are dealing with a CNF formula F, defined over variables $X = x_1, \ldots, x_n$, and we let $\mathcal{S}(F) \subseteq \{0,1\}^n$ denote the satisfying assignments of F.

DEFINITION 1. *The* diameter *of an arbitrary set $X \subseteq \{0,1\}^n$ is the largest Hamming distance between any two elements of X. The* distance *between two arbitrary sets $X, Y \subseteq \{0,1\}^n$ is the minimum Hamming distance between any $x \in X$ and any $y \in Y$. The* clusters *of a formula F are the connected components of $\mathcal{S}(F)$ when $x, y \in \{0,1\}^n$ are considered adjacent if they have Hamming distance* 1*. A* cluster-region *is a nonempty set of clusters.*

In [1] we proved that for all $k \geq 8$, for a range of densities below the satisfiability threshold, the set of satisfying assignments consists of exponentially many well-separated cluster-regions. Specifically, we proved the following theorem.

THEOREM 2 (see [1]). *For every $k \geq 8$, there exists a nonempty interval I_k and constants $\alpha_k < \beta_k < 1/2$ and $\epsilon_k > 0$ such that if $r \in I_k$, then w.h.p. the set of satisfying assignments of $F_k(n, rn)$ consists of $2^{\epsilon_k n}$ nonempty cluster-regions, such that*

1. *the diameter of each cluster-region is at most $\alpha_k n$,*
2. *the distance between any pair of cluster-regions is at least $\beta_k n$.*

If $r = (1-\delta)2^k \ln 2$, where $0 < \delta < 1/3$, then for all $k \geq k_0(\delta)$, we can take

$$\alpha_k = \frac{1}{k}, \qquad \beta_k = \frac{1}{2} - \frac{5}{6}\sqrt{\delta}, \qquad \epsilon_k = \frac{\delta}{2} - 3k^{-2}.$$

Here, our main result comes from "looking inside" clusters and proving the existence of variables which take the same value in all the truth assignments of a cluster. More formally, we have the following definition.

DEFINITION 3. *The* projection *of a variable x_i over a set of satisfying assignments C, denoted as $\pi_i(C)$, is the union of the values taken by x_i over the assignments in C. If $\pi_i(C) \neq \{0,1\}$, we say that x_i is* frozen *in C.*

The existence of frozen variables is a fundamental underpinning of the approximations implicit in the SP algorithm. A strength of our approach is that it allows us to prove not just the existence, but the pervasiveness of such variables. Specifically, Theorem 4 asserts that for a constant fraction of the satisfiable regime, the fraction of frozen variables in every single cluster is at least $1 - 2/k$. Thus, clusters shrink in volume and grow further apart by having smaller and smaller internal entropy (more frozen variables).

THEOREM 4. *For every $k \geq 9$, there exists $c_k < r_k$ such that for all $r \geq c_k$, w.h.p. every cluster of $F_k(n, rn)$ has at least $(1 - 2/k) \cdot n$ frozen variables. As k grows,*

$$\frac{c_k}{2^k \ln 2} \to \frac{4}{5}.$$

It remains open whether frozen variables exist for $k \leq 8$. As we mentioned above, [22] reported experimental evidence suggesting that frozen variables do *not* exist for $k = 3$.

We prove the existence of frozen variables by proving that, in fact, random formulas have nontrivial cores (we postpone the definition of cores and of the coarsening process until section 3). Thus, Theorem 9 answers affirmatively the main question of [22] for all $k \geq 9$. Combining Theorems 2 and 9, we get the following corollary, establishing that $F_k(n, rn)$ has an exponential number of distinct cores.

COROLLARY 5. *If $r = (1 - \delta)2^k \ln 2$, where $0 < \delta < 1/5$, then for all $k \geq k_0(\delta)$ w.h.p. every pair of truth assignments that belong in distinct cluster-regions have distinct coarsening fixed points.*

Proof. Since $\delta < 1/3$, Theorem 2 asserts that for all sufficiently large k, if σ, τ are truth assignments in distinct cluster regions, then w.h.p. their Hamming distance is at least $1/2 - (5/6)\sqrt{\delta}$. Since $\delta < 1/5$, Theorem 4 asserts that for all sufficiently large k, the fraction of $*$-variables in the coarsening fixed point of both σ and τ is bounded by $2/k$. It is easy to check that $1/2 - (5/6)\sqrt{\delta} > 2/k$ for all $\delta < 1/5$ and $k \geq 16$. Therefore, the coarsening fixed points of σ and τ must be distinct. □

3. Frozen variables: Survey propagation and related work. For a cluster C, the string $\pi(C) = \pi_1(C), \pi_2(C), \ldots, \pi_n(C)$ is the *projection* of C and we will use the convention $\{0, 1\} \equiv *$, so that $\pi(C) \in \{0, 1, *\}^n$. Imagine for a moment that given a formula F we could compute the marginal of each variable over the cluster projections, i.e., that for each variable we could compute the fraction of clusters in which its projection is $0, 1$, and $*$. Then, as long as we never assigned $\overline{x}$ to a variable which in every cluster was frozen to the value x, we are guaranteed to find a satisfying assignment: after each step there is at least one cluster consistent with our choices so far.

Being able to perform the above marginalization seems quite far fetched given that even if we are handed a truth assignment σ in a cluster C, it is not at all clear how to compute $\pi(C)$ in time independent of $|C|$. Survey Propagation (SP) is an attempt to compute marginals over cluster projections by making a number of approximations. One fundamental assumption underlying SP is that, unlike the marginals over truth assignments, the marginals over cluster projections essentially factorize, i.e., if two variables are far apart in the factor graph of the formula, then their joint distribution over cluster projections is essentially the product of their cluster projection marginals. Determining the validity of this assumption remains an outstanding open problem.

The other fundamental assumption underlying SP is that *approximate* cluster projections can be encoded as the solutions of a CSP whose factor graph can be syntactically derived from the input formula. Our results are closely related to this second assumption and establish that, indeed, the approximate cluster projections used in SP retain a significant amount of information from the cluster projections. To make this last notion concrete and enhance intuition, we give below a self-contained, brisk discussion of SP. For clarity of presentation this discussion is historically inaccurate. We attempt to restore history in section 3.1.

As we said above, even if we are given a satisfying assignment σ, it is not obvious

how to efficiently determine the projection of its cluster $C(\sigma)$. To get around this problem SP sacrifices information in the following manner.

DEFINITION 6. *Given a string $x \in \{0,1,*\}^n$, a variable x_i is* free *in x if in every clause c containing x_i or $\overline{x}_i$, at least one of the other literals in c is assigned true or $*$.*

We will refer to the following as a

coarsening step: *if a variable is free, assign it $*$.*

Given $x, y \in \{0,1,*\}^n$ say that x is dominated by y, written $x \preceq y$, if for every i, either $x_i = y_i$ or $y_i = *$.

Consider now the following process:

start at σ and apply coarsening until a fixed point is reached.

LEMMA 7. *For every formula F and truth assignment $\sigma \in \mathcal{S}(F)$, there is a unique coarsening fixed point $w(\sigma)$. If σ_1, σ_2 belong to the same cluster C, then $w(\sigma_1) = w(\sigma_2) \succeq \pi(C)$.*

Proof. Trivially, applying a coarsening step to a string x produces a string y such that $x \preceq y$. Moreover, if x_i was free in x, then y_i will be free in y. As a result, if both $y, z \in \{0,1,*\}^n$ are reachable from $x \in \{0,1,*\}^n$ by coarsening steps, so is the string that results by starting at x, concatenating the two sequences of operations and removing all but the first occurrence of each coarsening step. This implies that there is a unique fixed point $w(x)$ for each $x \in \{0,1,*\}^n$ under coarsening. Observe now that if $\sigma, \sigma' \in \mathcal{S}(F)$ differ only in the ith coordinate, then the ith variable is free in both σ, σ' and coarsening it in both yields the same string τ. By our earlier argument, $w(\sigma) = w(\tau) = w(\sigma') = w_C$, where $C \subseteq \mathcal{S}(F)$ is the cluster containing σ, σ'. Considering all adjacent pairs in C, we see that $w_C \succeq \pi(C)$. ∎

DEFINITION 8. *The* core *of a cluster C is the unique coarsening fixed point of the truth assignments in C.*

By Lemma 7, if a variable takes either the value 0 or the value 1 in the core of a cluster C, then it is frozen to that value in C. To prove Theorem 4 we prove that the core of every cluster has many non-$*$ variables.

THEOREM 9. *For every $k \geq 9$, there exists $c_k < r_k$ such that for all $r \geq c_k$, w.h.p. the coarsening fixed point of* every *$\sigma \in \mathcal{S}(F_k(n, rn))$ contains fewer than $(2/k) \cdot n$ variables that take the value $*$.*

To prove Theorem 9 (which implies Theorem 4) we derive sharp bounds for the large deviations rate function of the coarsening process applied to a fixed satisfying assignment. As a result, we also prove that in the planted assignment model the cluster containing the planted assignment already contains frozen variables at $r \sim (2^k/k) \ln k$. Also, we will see that our proof gives a strong hint that for small values of k, such as $k = 3$, for all densities in the satisfiable regime, most satisfying assignments *do* converge to $(*, \ldots, *)$ upon coarsening.

We can think of coarsening as an attempt to estimate the projection of $C(\sigma)$ by starting at σ and being somewhat reckless. To see this, consider a parallel version of coarsening in which, given $x \in \{0,1,*\}^n$, we coarsen all free variables in it simultaneously. Clearly, the first round of this process applied to σ will only assign $*$ to variables whose projection in $C(\sigma)$ is indeed $*$. Subsequent rounds, though, might not: a variable v is deemed free if in every clause containing it there is some other variable satisfying the clause, *or* a variable assigned $*$. This second possibility is equivalent to assuming that the $*$-variables in the clauses containing v, call them Γ_v, can take joint values that allow v to not contribute in the satisfaction of any clause. In general

formulas this is, of course, not a valid assumption. On the other hand, the belief that in random formulas there are no long-range correlations *among the nonfrozen* variables of each cluster makes this a reasonable statistical assumption: since the formula is random, the variables in Γ_v are probably far apart from one another in the factor graph that results after removing the clauses containing v. Thus, indeed, any subset of variables of Γ_v that do not co-occur in a clause should be able to take *any* set of joint values. Our results can be seen as evidence of the utility of this line of reasoning, since we prove that for sufficiently large densities, the coarsening fixed point of a satisfying assignment is *never* $(*, \ldots, *)$. Indeed, as we approach the satisfiability threshold, the fraction of frozen variables in it tends to 1.

Of course, while the core of a cluster C can be easily derived given some $\sigma \in C$, such a σ is still hard to come by. The last leap of approximation underlying SP is to define a set $Z(F) \subseteq \{0, 1, *\}^n$ that includes all cluster cores, yet is such that membership in $Z(F)$ is "locally checkable," akin to membership in $\mathcal{S}(F)$. Specifically, we present the following definition.

DEFINITION 10. *A string $x \in \{0, 1, *\}^n$ is a* cover *of a CNF formula F if* (i) *under x, every clause in F contains a satisfied literal or at least two $*$, and* (ii) *every free variable in x is assigned $*$, i.e., x is $*$-maximal.*

Cores trivially satisfy (ii) as fixed points of coarsening. It is also easy to see that any string that results by applying coarsening steps to a satisfying assignment satisfies (i): the unique satisfying literal of a clause will not become $*$ unless the clause also has other $*$ variables. Thus, a core is always a cover. At the same time, checking whether $x \in \{0, 1, *\}^n$ satisfies (i) can be done trivially by examining each clause in isolation. For (ii) it is enough to check that for each variable v assigned 0 or 1 in x, there is at least one clause satisfied by v and dissatisfied by all other variables in it. Again, this amounts to n simple checks, each check done in isolation by considering the clauses containing the corresponding variable. The price we pay for dealing with locally checkable objects is that the set of all covers $Z(F)$ can be potentially much bigger than the set of all cores. For example, $(*, \ldots, *)$ is always a cover, even if F is unsatisfiable.

The SP algorithm can now be stated as follows.

Repeat until all variables are set:

1. Compute the marginals of variables over covers.
2. Select a variable with least probability mass on $*$ and assign it the 0/1 value on which it puts most mass.
3. Simplify the formula.

The computation of marginals over covers in the original derivation [23] of SP was, in fact, done via a message passing procedure that runs on the factor graph of the original formula rather than a factor graph encoding covers (more on this in section 3.1). Also, in [23], if a configuration is reached in which all variables put (nearly) all their mass on $*$, the loop is stopped and a local search algorithm is invoked. The idea is that when such a configuration is reached, the algorithm has "arrived" at a cluster and finding a solution inside that cluster is easy since only nonfrozen variables remain unset.

3.1. Related work. The original presentation of SP motivated the algorithm in terms of a number of physical notions (cavities, magnetic fields, etc.). Specifically, the algorithm was derived by applying the "cavity method" within a "1-step replica symmetry breaking" scheme, with no reference whatsoever to notions such as cluster projections, cores, or covers. On the other hand, a very definitive message passing

procedure was specified on the factor graph of the original formula, and the computer code accompanying the paper and implementing that procedure worked spectacularly well. Moreover, a notion foreshadowing cores was included in the authors' discussion of "Warning Propagation."

Casting SP as an attempt to compute marginals over cores was done independently by Braunstein and Zecchina in [10] and Maneva, Mossel, and Wainwright in [22]. In particular, in both papers it is shown that the messages exchanged by SP over the factor graph of the input formula are the messages implied by the Belief Propagation formalism [8] applied to a factor graph encoding the set of all covers. In [22], the authors also established a number of formal correspondences between SP, Markov random fields, and Gibbs sampling. In particular, they noted that a cover $\sigma \in \{0,1,*\}^n$ can also be thought of as partial truth assignment in which every unsatisfied clause has length at least 2, and in which every variable v assigned 0 or 1 has some clause c for which it is essential in σ, i.e., v satisfies c but all other variables in c are set opposite to their sign in c. This last view motivates a generalization of SP in which marginals are computed not only over covers, but over all partial assignments in which every unsatisfied clause has length at least 2, weighted exponentially in the number of nonessential 0/1 variables and the number of $*$-variables. One particular motivation for this generalization is that while SP appears to work very well on random 3-CNF formulas, [22] gives experimental evidence that such formulas do not have nontrivial cores, i.e., upon coarsening, truth assignments end up as $(*,\ldots,*)$. This apparent contradiction is reconciled by attributing the success of SP to the existence of "near-core" strings allowed under the proposed generalization. It is also worth keeping in mind that in the experiments of [22], the truth assignments to which coarsening is applied are obtained by running SP plus decimation, a process which most likely introduces significant bias in the choice among satisfying assignments.

While [22] provided a framework for studying SP by connecting it to concrete mathematical objects such as cores and Markov random fields, it did not provide results on the actual structure of the solution space of random k-CNF formulas. Indeed, motivated by the experimental absence of cores for $k = 3$, the authors asked whether random formulas have nontrivial cores for any k. Our results establish a positive answer to this question for all $k \geq 9$.

4. The probabilistic framework. Theorem 4 follows from Theorem 9 and Lemma 7. To prove Theorem 9 we say that a satisfying assignment σ is α-coreless if its coarsening fixed point $w(\sigma)$ has at least αn $*$-variables. Let X be the random variable equal to the number of α-coreless satisfying assignments in a random k-CNF formula $F_k(n, rn)$. By symmetry, writing $0^n = \mathbf{0}$,

$$\mathbb{E}[X] = \sum_{\sigma\in\{0,1\}^n} \Pr[\sigma \text{ is } \alpha\text{-coreless} \mid \sigma \text{ is satisfying}] \cdot \Pr[\sigma \text{ is satisfying}] \tag{2}$$

$$= 2^n \cdot \left(1 - \frac{1}{2^k}\right)^{rn} \cdot \Pr[\mathbf{0} \text{ is } \alpha\text{-coreless} \mid \mathbf{0} \text{ is satisfying}]. \tag{3}$$

Observe that conditioning on "$\mathbf{0}$ is satisfying" is exactly the same as "planting" the $\mathbf{0}$ solution, and amounts to selecting the $m = rn$ random clauses in our formula, uniformly and independently from amongst all clauses having at least one negative literal. We will see that for every $k \geq 3$, there exists t_k^α such that

$$\Pr[\mathbf{0} \text{ is } \alpha\text{-coreless} \mid \mathbf{0} \text{ is satisfying}] = \begin{cases} 1 - o(1) & \text{if } r < t_k^\alpha, \\ o(1) & \text{if } r > t_k^\alpha. \end{cases} \tag{4}$$

In particular, we will see that $t_k^1 \sim (2^k/k)\ln k$. We find it interesting (and speculate it is not an accident) that all algorithms that have been analyzed so far work for densities below t_k^1. More precisely, all algorithms analyzed so far set each variable v by considering only a subset of the not-yet-satisfied clauses containing v and succeed for some $r < c\,2^k/k$, where c depends on the algorithm.

To prove $\mathbb{E}[X] = o(1)$ we derive a strong upper bound for the probability in (4) when $r \gg t_k^\alpha$. Specifically, we will prove that $\Pr[\mathbf{0}$ is α-coreless $\mid \mathbf{0}$ is satisfying$] < e^{-f(r)n}$ for a function f such that for all $r \geq c_k^\alpha$,

$$2 \cdot \left(1 - \frac{1}{2^k}\right)^r \cdot e^{-f(r)} < 1. \tag{5}$$

By (3), for all such r we have $\mathbb{E}[X] = o(1)$ and Theorem 9 follows.

4.1. Coarsening as hypergraph stripping. Given any CNF formula F and any $\sigma \in \mathcal{S}(F)$ it is easy to see that $w(\sigma)$ is completely determined by the set of clauses $U(\sigma)$ that have precisely one satisfied literal under σ. This is because after any sequence of coarsening steps applied to σ, a clause that had two or more satisfied literals under σ will have at least one satisfied literal or at least two $*$ and thus never prevent any variable from being free. Therefore, to coarsen a truth assignment σ it is enough to consider the clauses in $U(\sigma)$. Let us say that a variable v is unfrozen if there is no clause in which it is the unique satisfying variable, and let us say that a clause is unfrozen if it contains an unfrozen variable (and frozen otherwise). It is now easy to see that coarsening σ is equivalent to starting with U and removing unfrozen clauses, one by one, in an arbitrary order until a fixed point is reached, i.e., no unfrozen clauses remain. Variables occurring in any remaining (frozen) clauses are, thus, frozen in $w(\sigma)$ (to their value in σ), while all other variables are assigned $*$. This view of coarsening as repeated removal of clauses from $U(\sigma)$ will be very useful in our probabilistic analysis below.

To estimate $\Pr[\mathbf{0}$ is α-coreless $\mid \mathbf{0}$ is satisfying$]$ we consider a random k-CNF formula with rn clauses chosen uniformly among those satisfying $\mathbf{0}$. To determine $w(\mathbf{0})$, by our discussion above, it suffices to consider the clauses in our formula that have precisely one satisfied (negative) literal. The number of such clauses is distributed as

$$m = \mathrm{Bin}\left(rn, \frac{k}{2^k - 1}\right).$$

It will be convenient to work in a model where each of these m clauses is formed by choosing 1 negative literal and $k-1$ positive literals, uniformly, independently, *and with replacement*. (Since $m = O(n)$, by standard arguments, our results then apply when replacement is not allowed and the original number of clauses is $rn - o(n)$.) We think of the k literals in each clause as k balls; we paint the single satisfied literal of each clause red, and the $k-1$ unsatisfied literals blue. We also have one bin for each of the n variables, and we place each literal in the bin of its underlying variable. We will use the term "blue bin" to refer to a bin that has at least one blue ball and no red balls. With this picture in mind, we see that the $*$-variables in $w(\mathbf{0})$ correspond precisely to the set of empty bins when the following process terminates:

1. Let v be any blue bin; if none exists exit.
   %Identify an unfrozen variable v if one exists.
2. Remove any ball from v.
   %Remove the occurrence of v in some (unfrozen) clause c.

3. Remove $k-2$ random blue balls.
   %Remove the other $k-2$ unsatisfied literals of c.
4. Remove a random red ball.
   %Remove the satisfied literal in c.

Note that the above process removes exactly one clause (1 red ball and $k-1$ blue balls) in each step and, therefore, if we pass the condition in step 1, there are always suitable balls to remove. To give a lower bound on the probability that the process exits before m steps (thus, reaching a nontrivial fixed point), we will give a lower bound on the probability that it exits within the first $i = \alpha m$ steps, for some carefully chosen $\alpha = \alpha(k, r) \in (0, 1)$. In particular, observe that for the process to not exit within the first i steps the following must be true:

(6) At the beginning of each of the first i steps there is at least one blue bin.

To bound the probability of the event in (6) we will bound the probability it occurs in the following simplified process. The idea is that this modified process is significantly easier to analyze, while the event in (6) is only slightly more likely, at least for the values of k, r of interest to us.

(a) Let v be any blue bin; if none exists go to step (c).
(b) Remove any ball from v.
(c) Remove a random red ball.

LEMMA 11. *The event in* (6) *is no less likely in the modified process than in the original process.*

We prove Lemma 11 below. To bound the probability of the event in (6) in the modified process we argue as follows. Let q be the number of bins which do not contain any red ball after i steps, and let b be the original number of blue balls in these q bins. If $b < i$, then after b steps of the modified process every nonempty bin will contain at least one red ball, since up to that point we remove precisely one blue ball per step. Therefore, the probability of the event in (6) is bounded above by the probability that $b \geq i$. To bound this last probability we observe that the red balls in the modified process evolve completely independently of the blue balls. Moreover, since we remove exactly one red ball in each step, the state of the red balls after i steps is distributed exactly as if we had simply thrown $m - i$ red balls into the n bins.

So, all in all, given a random k-CNF formula F with rn clauses and a fixed $0 \leq i \leq n$, conditional on $\mathbf{0}$ satisfying F, the probability that the coarsening process started at $\mathbf{0}$ fails to reach a fixed point within i steps is bounded by the probability that $b \geq i$, where

$$b = \mathrm{Bin}\left((k-1)m, \frac{q}{n}\right), \quad \text{where} \tag{7}$$

$$m = \mathrm{Bin}\left(rn, \frac{k}{2^k - 1}\right), \quad \text{and} \tag{8}$$

$$q = \mathrm{Emp}\left(m - i, n\right), \tag{9}$$

where $\mathrm{Emp}(x, y)$ is the distribution of the number of empty bins when we throw x balls into y bins.

As a result, given k, r, our goal is to determine a value for i that minimizes $\Pr[b \geq i]$. Before we delve into the probabilistic calculations, in the next section we comment on how our analysis relates to the planted assignment problem and to the existence of nontrivial cores for small values of k.

Proof of Lemma 11. Consider a process which is identical to the original process except with step 3 removed. We will call this the intermediate process. We begin by proving that the original and intermediate processes can be coupled so that whenever the event in (6) occurs in the original process it also occurs in the intermediate process.

First, observe that the evolution of the red balls in both processes is purely random and therefore can be assumed to be identical, i.e., we can think of the original process as making a genuine random choice in step 4 and the intermediate process as mimicking that choice. (We think of all balls as carrying a distinct identifier.) Similarly, we can assume that originally the placement of the blue balls in bins is identical for the two processes.

Let us say that a pair of blue ball placements is good if in every bin the set of blue balls in the original process is a subset of the set of blue balls in the intermediate process. Clearly, whenever we are in a good configuration, since the placement of the red balls is identical in the two processes, any choice of bin and ball of the original process in steps 1 and 2 is an available choice for the intermediate process. Moreover, if the intermediate process mimics these choices, this results in a new good pair of blue ball placements. By induction, since the original pair of blue ball placements is good, if the event in (6) occurs in the original process, it also occurs in the intermediate process.

Next, we compare the intermediate process to the modified process observing that they are identical except that in the event that we run out of bins containing only blue balls the intermediate process stops, while the modified process carries on. Therefore, we couple the two as follows: the modified process mimics the intermediate process for as long as the event in (6) does not occur, and makes its own random choices afterwards. Therefore, if the event in (6) occurs in the intermediate process, it also occurs in the modified process. ∎

5. The planted assignment model and small values of k. Conditional on $\mathbf{0}$ being satisfying, analyzing $w(\mathbf{0})$ is exactly the same as working in the "planted assignment" model and analyzing the core of the cluster containing the planted assignment. This is rather easy to do if we are content with results holding with probability $1-o(1)$. Specifically, by (7), (8), (9), and standard concentration results it follows immediately that if $i = \alpha m$, then w.h.p.

$$m = \lambda \cdot n + o(n), \text{ where } \lambda = \frac{rk}{2^k - 1}, \tag{10}$$

$$q = \gamma \cdot n + o(n), \text{ where } \gamma = \exp\left(-\lambda(1-\alpha)\right), \tag{11}$$

$$b = \beta \cdot n + o(n), \text{ where } \beta = (k-1)\gamma\lambda. \tag{12}$$

With these conditionals in place, we can next determine the mean path of the coarsening process using the method of differential equations [24], i.e., the number of red and blue balls after each step, up to $o(n)$. In particular, this allows us to show the following claim.

Claim 12. *For every $k \geq 3$, there exists a critical value t_k^1 such that if $r < t_k^1$, then w.h.p. $w(\mathbf{0}) = (*, \ldots, *)$, while if $r > t_k^1$, then w.h.p. a bounded fraction of the variables in $w(\mathbf{0})$, and therefore in $C(\mathbf{0})$, are frozen.*

In Table 2 we give the value of t_k^1 for some small values of k (rounding to two decimals). We see that for $k = 3, 4, 5$, the probability that $\mathbf{0}$ has a nontrivial coarsening fixed point conditional on being satisfying tends to 0 for all densities in the satisfiable regime. Clearly, conditioning on "$\mathbf{0}$ is satisfying" is not the same as picking a "typical"

satisfying assignment. Nevertheless, the gap between t_k^1 and the best threshold upper bound for $k = 3$ is sufficiently large to suggest that most satisfying assignments do arrive at $(*, \ldots, *)$ upon coarsening. This is consistent with the experimental results of [22], who first raised this possibility. That said, a distinction worth mentioning is that even if the coarsening procedure arrives at $(*, \ldots, *)$ from most/all satisfying assignments there can still be (many) frozen variables: simply, their corresponding clusters may not be compact ("cube-like") enough for coarsening to discover their core.

TABLE 2

k	3	4	5	6	7
Best known lower bound for r_k	3.52	7.91	18.79	40.62	84.82
Best known upper bound for r_k^*	4.51	10.23	21.33	43.51	87.88
Nontrivial coarsening fixed point for $\mathbf{0}$, t_k^1	5.72	11.58	21.75	40.13	73.88
Threshold for the modified process, u_k	6.25	12.34	22.90	41.95	76.84

We now comment on the couple of simplifications of the original process that we introduced in the previous section in order to get a process that is easier to analyze. As we showed, these simplifications only increase the probability of the event in (6). It is natural to wonder if this increase is significant, allowing for the possibility that our analysis can be made much sharper. Below we give evidence that this is not the case. In particular, if each of m, q, b can be assumed to be within $o(n)$ of its expected value, then the inequality $b \geq i$ in the modified process is equivalent to

$$r < \frac{2^k - 1}{k} \cdot \frac{\ln\left(\frac{k-1}{\alpha}\right)}{1-\alpha} \equiv u_k(\alpha).$$

In Table 2 we give the value of $u_k = \min_\alpha u_k(\alpha)$ for some small values of k. As we can see, these values are quite close to t_k^1 and get relatively closer as k is increased. In other words, considering the modified process does not cause too big a loss in the analysis. Indeed, taking $\alpha = 1/\ln k$, already gives $u_k \to (2^k/k)\ln k$, which is consistent with the physics prediction that $t_k^1 \to (2^k/k)\ln k$.

Of course, if one is interested in establishing that certain properties of $w(\mathbf{0})$ hold with exponentially small failure probability, as we do, then conditioning that m, q, b are within $o(n)$ of their expectation is not an option. One has to do a large deviation analysis of all these variables and their interactions in the coarsening process and determine the dominant source of fluctuations. This is precisely what we do with respect to the event $b \geq i$ in the modified process.

6. Large deviations. It is well known that if $np > 0$, then for every $\delta \geq -1$,

$$\Pr[\mathrm{Bin}(n,p) = (1+\delta)np] \leq F(np, \delta),$$

where

$$F(x,y) = \exp(-x[(1+y)\ln(1+y) - y]).$$

A similar large deviations bound was shown in [17] for the number of empty bins in a balls-in-bins experiment (Theorem 3). That is, for every $\delta \geq -1$,

$$\Pr[\mathrm{Emp}(m,n) = (1+\delta)e^{-m/n}] \leq F(ne^{-m/n}, \delta).$$

6.1. Application. Write $r = \lambda(2^k - 1)/k$ and fix $\delta, \epsilon, \zeta \geq -1$. Write $\rho = \lambda(1 + \delta)(1 - \alpha)$ in order to compress the expressions below. The probability that

$$m = (1+\delta)\mathbb{E}[m] = (1+\delta)\frac{rk}{2^k - 1} \cdot n = (1+\delta)\lambda \cdot n, \tag{13}$$

$$q = (1+\zeta)\mathbb{E}[q|m] = (1+\zeta)\exp\left(-\frac{m-i}{n}\right) \cdot n = (1+\zeta)e^{-\rho} \cdot n, \tag{14}$$

$$\begin{aligned} b &= (1+\epsilon)\mathbb{E}[b|q,m] = (1+\epsilon)(k-1)m \cdot \frac{q}{n} \\ &= (1+\delta)(1+\epsilon)(1+\zeta)\lambda(k-1)e^{-\rho} \cdot n \end{aligned} \tag{15}$$

is bounded by

$$F(\lambda n, \delta) \cdot F(e^{-\rho}n, \zeta) \cdot F((1+\delta)(1+\zeta)\lambda(k-1)e^{-\rho}n, \epsilon). \tag{16}$$

We write this as $e^{-n\,\Omega}$, where

$$\Omega \equiv \lambda\omega(\delta) + e^{-\rho}\omega(\zeta) + \lambda(k-1)(1+\delta)(1+\zeta)e^{-\rho}\omega(\epsilon), \tag{17}$$

with $\omega(x) = (1+x)\ln(1+x) - x$.

Conditional on the events in (13)–(15) we see from (15) that the condition $b \geq i$ becomes $B \geq 0$, where

$$B \equiv (1+\epsilon)(1+\zeta)(k-1)e^{-\rho} - \alpha.$$

For any fixed k, r, and α, define $\Phi \equiv \{(\delta, \zeta, \epsilon) : B \geq 0\}$. Thus,

$$\Pr[\mathbf{0} \text{ is } \alpha\text{-coreless} \mid \mathbf{0} \text{ is satisfying}] < \exp(-n \cdot \min_{\Phi} \Omega) \times \mathrm{poly}(n),$$

and to prove that the expected number of α-coreless assignments in $o(1)$, it suffices to prove

$$\min_{\Phi} \Omega > \ln 2 + r\ln(1 - 2^{-k}) \equiv s. \tag{18}$$

7. Optimization. To establish (18) it is enough to prove that the maximum of B in the variables δ, ζ, and ϵ under the condition $\Omega \leq s$ is negative. Considering that the function B is monotone in the three variables δ, ζ, and ϵ, the maximizer of B in the region $\Omega \leq s$ has to be on the boundary, that is, for $\Omega = s$. The maximum of B under the condition $\Omega = s$ corresponds to the extremum of the function $G = B - \mu(\Omega - s)$, where μ is a Lagrange multiplier. The equations for the location of the maximizer are thus given by derivatives of G with respect to δ, ζ, ϵ, and μ,

$$\partial_\delta G = 0 \quad \Rightarrow \quad \partial_\delta B = \mu\, \partial_\delta \Omega, \tag{19}$$

$$\partial_\zeta G = 0 \quad \Rightarrow \quad \partial_\zeta B = \mu\, \partial_\zeta \Omega, \tag{20}$$

$$\partial_\epsilon G = 0 \quad \Rightarrow \quad \partial_\epsilon B = \mu\, \partial_\epsilon \Omega, \tag{21}$$

$$\partial_\mu G = 0 \quad \Rightarrow \quad \Omega = s. \tag{22}$$

LEMMA 13. *For any fixed k, r, and $\alpha \in (0,1)$, at the extremum of the function G defined by* (19)–(22) *the following assertions hold:*

1. *ϵ is nonnegative.*
2. *ζ is nonnegative.*

3. *δ is nonpositive.*

Proof. The first assertion follows by observing that B is an increasing function of ϵ and Ω contains ϵ only in the third term through $\omega(\epsilon)$. Therefore, if the maximizer would be in $\epsilon = \epsilon' < 0$, moving to $\epsilon = \epsilon'' > 0$, with $\omega(\epsilon') = \omega(\epsilon'')$, would keep Ω constant while increasing B.

Combining (20) and (21) in order to remove μ, we have

$$\partial_\zeta \Omega = \partial_\epsilon \Omega \, \partial_\zeta B / \partial_\epsilon B,$$

that is,

$$e^{-\rho} \ln(1+\zeta) + (1+\delta)\lambda(k-1)e^{-\rho}\omega(\epsilon) = (1+\delta)\lambda(k-1)e^{-\rho}(1+\epsilon)\ln(1+\epsilon),$$

which, after simplification, reduces to

$$\ln(1+\zeta) = (k-1)\lambda(1+\delta)\epsilon. \tag{23}$$

Thus, for $\delta \geq -1$ and $\epsilon \geq 0$ we have that at the maximizer $\zeta \geq 0$, proving the second assertion.

Combining (19) and (21) we can write

$$\begin{aligned} 0 = \partial_\delta \Omega - \partial_\epsilon \Omega \, \partial_\delta B / \partial_\epsilon B = \lambda e^{-\rho} \Big[& e^{\rho} \ln(1+\delta) - (1-\alpha)\omega(\zeta) - \rho(1+\zeta)(k-1)\omega(\epsilon) \\ & + (k-1)(1+\zeta)\omega(\epsilon) + \rho(1+\epsilon)(1+\zeta)(k-1)\ln(1+\epsilon) \Big]. \end{aligned}$$

The term within square brackets can be simplified to

$$e^{\rho} \ln(1+\delta) - (1-\alpha)\omega(\zeta) + \rho(1+\zeta)(k-1)\epsilon + (k-1)(1+\zeta)\omega(\epsilon),$$

which, using (23), implies

$$e^{\rho} \ln(1+\delta) + (1-\alpha)\zeta + (k-1)(1+\zeta)\omega(\epsilon) = 0.$$

Since for $\zeta, \epsilon \geq 0$ the second and third terms of this expression are nonnegative, we find that δ has to be nonpositive at the maximizer in order to satisfy the last equation (third assertion). ∎

We next prove some bounds on δ and ϵ that hold at the maximizer.

LEMMA 14. *Fix any r, k, and $\alpha \in (0,1)$. At the maximizer of B,*

$$\delta_0 \equiv -\sqrt{\frac{2s}{\lambda}} \leq \delta \leq 0.$$

Proof. Since δ in nonpositive at the maximizer, we observe that $\omega(\delta) \geq \delta^2/2$ for $\delta \leq 0$. Moreover, each of the three terms in Ω is nonnegative for $\delta, \zeta \geq -1$ and this implies

$$\lambda\omega(\delta) \leq s \implies \lambda\frac{\delta^2}{2} \leq s \implies |\delta| \leq \sqrt{\frac{2s}{\lambda}}. \qquad \blacksquare$$

LEMMA 15. *Fix any r, k, and $\alpha \in (0,1)$. At the maximizer of B,*

$$\epsilon < \frac{1-\alpha}{k-1} + \frac{\ln 3}{\lambda(1+\delta_0)(k-1)} \equiv \epsilon_0.$$

Proof. Since every term in Ω is nonnegative, considering the second term and using the fact $s = \Omega$ we get

$$s \geq \omega(\zeta)e^{-\rho}.$$

Observe now that for all $x \geq -1$, we have $\omega(x) > x - 1$. Therefore,

$$se^{\rho} + 2 > 1 + \zeta. \tag{24}$$

Using (23) to replace $1 + \zeta$ and the fact $s \leq \ln 2 < 1$ we can conclude from (24) that

$$e^{\rho} + 2 > e^{\lambda(1+\delta)(k-1)\epsilon} \implies \epsilon < \frac{\ln(e^{\rho} + 2)}{\lambda(1+\delta)(k-1)} \leq \frac{\rho + \ln 3}{\lambda(1+\delta)(k-1)},$$

where for the last inequality we use that $\rho = \lambda(1 + \delta)(1 - \alpha)$ is nonnegative. We conclude that at the maximizer

$$\epsilon < \frac{1-\alpha}{k-1} + \frac{\ln 3}{\lambda(1+\delta_0)(k-1)} \equiv \epsilon_0,$$

where δ has been replaced by its lower bound value. ∎

Thus, the stationary point of G must occur in the region $\Lambda = \{(\delta, \epsilon) : \delta_0 \leq \delta \leq 0, 0 \leq \epsilon \leq \epsilon_0\}$. In the next subsections we derive analytical results for this optimization for all $k \geq 14$, and we summarize results obtained by numerically finding the stationary point of G for $9 \leq k \leq 13$.

7.1. Proving the existence of frozen clauses for $k \geq 14$ analytically. For any fixed values of δ and ϵ, the requirement $B \geq 0$ implies

$$\zeta \geq \frac{\alpha\, e^{\rho}}{(k-1)(1+\epsilon)} - 1. \tag{25}$$

Plugging this lower bound in the second term of Ω, we see that the requirement $\Omega = s$ implies

$$\frac{\alpha}{(k-1)(1+\epsilon)} \left[\ln\left(\frac{\alpha\, e^{\rho}}{(k-1)(1+\epsilon)} \right) - 1 \right] + e^{-\rho} \leq s. \tag{26}$$

Therefore, it suffices to find λ and α such that (26) cannot be satisfied by any $\delta_0 \leq \delta \leq 0$ and $0 \leq \epsilon \leq \epsilon_0$. This is certainly true if a lower bound to the left-hand side of (26) makes such an equation unsatisfied, that is, if

$$\frac{\alpha}{(k-1)(1+\epsilon_0)} \left[\ln\left(\frac{\alpha\, e^{\lambda(1+\delta_0)(1-\alpha)}}{(k-1)(1+\epsilon_0)} \right) - 1 \right] + e^{-\lambda(1-\alpha)} > s$$

and the term within the squared brackets above is positive. Thus, to summarize, it suffices to find λ and α such that

$$\frac{\alpha}{(k-1)(1+\epsilon_0)} \left[\ln\left(\frac{\alpha\, e^{\lambda(1+\delta_0)(1-\alpha)}}{(k-1)(1+\epsilon_0)} \right) - 1 \right] > s - e^{-\lambda(1-\alpha)} > 0. \tag{27}$$

With the change of variable $\lambda = c\, k \ln 2$ we have that

$$s = \ln 2 + c(2^k - 1)\ln(1 - 2^{-k}) \ln 2 \leq \left(1 - c(1 - 2^{-k})\right) \ln 2$$

and

$$s - e^{-\lambda(1-\alpha)} \leq \left(1 - c(1-2^{-k})\right) \ln 2 - 2^{-ck(1-\alpha)} < (1-c)\ln 2,$$

since for any $c \in [0,1]$, $\alpha \in [0,1]$, and $k > 0$,

$$c \ln 2 < 2^{k(1-c(1-\alpha))}.$$

Therefore, it suffices to establish

$$\frac{\alpha}{(k-1)(1+\epsilon_0)} \left[\ln \left(\frac{\alpha\, e^{ck(1+\delta_0)(1-\alpha)\ln 2}}{(k-1)(1+\epsilon_0)} \right) - 1 \right] \geq (1-c)\ln 2. \tag{28}$$

Based on Lemmas 14 and 15 we now introduce simpler bounds for δ and ϵ, which hold for all $c \geq 4/5$ and $k \geq 2$. Specifically,

$$|\delta_0| \leq \sqrt{\frac{2(1-c(1-2^{-k}))}{c\,k}} \leq \sqrt{\frac{2(1-3/4\,c)}{c\,k}} \leq \frac{1}{\sqrt{k}} \tag{29}$$

and

$$\epsilon_0 = \frac{1}{k-1}\left(1 - \alpha + \frac{\ln 3}{c\,k(1+\delta_0)\ln 2}\right) \leq \frac{2}{k-1}. \tag{30}$$

Replacing (29) and (30) in (28) we have

$$\frac{\alpha}{k+1}\left[c\,k\,(1-\alpha)\left(1 - \frac{1}{\sqrt{k}}\right)\ln 2 + \ln\left(\frac{\alpha}{k+1}\right) - 1\right] - (1-c)\ln 2 \geq 0.$$

Solving with respect to c, the last inequality becomes

$$c \geq \frac{1 + \frac{\alpha}{k+1}\left[1 - \ln\left(\frac{\alpha}{k+1}\right)\right]/\ln 2}{1 + \alpha(1-\alpha)\frac{1-1/\sqrt{k}}{1+1/k}} \equiv g_c(k,\alpha).$$

For any fixed $\alpha \in (0,1)$, $g_c(k,\alpha)$ is a decreasing function of k, which as $k \to \infty$ tends to

$$\frac{1}{1+\alpha(1-\alpha)}.$$

In order to prove that there exists a choice of α such that $\min_\Phi \Omega > s$ for some $r < r_k$ and all $k \geq k_0(\alpha)$, we rescale the lower bound for r_k from (1) as

$$\tau_k \equiv \frac{2^k \ln 2 - \frac{(k+1)\ln 2 + 3}{2}}{(2^k - 1)\ln 2} = \frac{1}{1-2^{-k}} - \frac{(k+1)\ln 2 + 3}{2(2^k-1)\ln 2}$$

and observe that τ_k is increasing in k. In Figure 1 we now see that the function $g_c(14,\alpha)$ dips below τ_{14} for a certain range of α, implying that the left endpoint of this range is an upper bound on the fraction of unfrozen clauses. For larger values of k things only get better since $g_c(k,\alpha)$ is monotonically decreasing with k, whereas τ_k is increasing. For any fixed value of α, $k_0(\alpha)$ can be defined as the first k value for which $g_c(k,\alpha) < \tau_k$ holds.

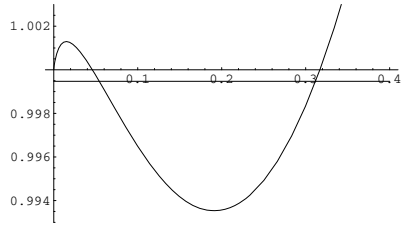

FIG. 1. *The function* $g_c(14, \alpha)$. *The horizontal line, slightly below* 1, *is* $\tau_{14} = 0.999471....$

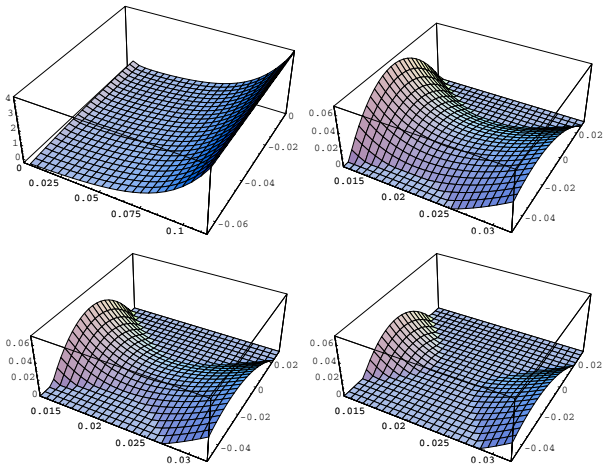

FIG. 2. $k = 9$, $\alpha = 0.265$.

7.2. The $k = 9$ case. Recall that for any fixed k, r, and α the function G depends on four variables: δ, ϵ, ζ, and μ. We will plot G for $k = 9$, $\alpha = 0.265$, and a few different values of r, while fixing ζ and μ at the value they take at the stationary point: ζ is given readily by (23), and substituting this value of ζ into (21) we get

$$\mu = \frac{1}{\lambda(1+\delta)\ln(1+\epsilon)}.$$

In the upper left panel of Figure 2 we show G in the subregion of Λ corresponding to the optimal ζ, μ for $r = 347$. By closer inspection one finds that there is a unique stationary point in this region. The remaining three plots are zoomed on the stationary point for $r = 347$, $r = 347.5$, and $r = 348$. It is clear that the function G at the stationary point is positive for the first two r values and negative for the third one (for the sake of clarity, negative values of G are not plotted). Thus, for $k = 9$ and $\alpha = 0.265$, the critical value of r lies between 347.5 and 348. In the next subsection we determine this critical value numerically for all $9 \le k \le 13$.

7.3. Optimizing for $9 \le k \le 13$. For any fixed k and α the value of c_k^α, such that w.h.p. clustering exists for $r > c_k^\alpha$, can be computed by solving numerically (19)–(22) together with $G = 0$ (which reduces to $B = 0$ since $\Omega = s$ by (22)). Adding a sixth equation $\partial_\alpha G = 0$ allows one to minimize c_k^α with respect to α (at some α_m) thus determining the smallest density c_k for which the existence of frozen variables can be established. Numerical solutions of these six equations are given in Table 3 for

TABLE 3

k	r_k	c_k	α_m	μ	δ	ζ	ϵ
9	349.92	347.84	0.265	8.037	-0.015085	1.7336	0.02083
10	704.94	690.48	0.273	6.935	-0.015714	2.7134	0.02194
11	1413.90	1370.42	0.281	6.256	-0.015789	4.0330	0.02229
12	2833.12	2720.44	0.289	5.802	-0.015548	5.7977	0.02220
13	5671.90	5402.23	0.297	5.480	-0.015132	8.1457	0.02184

$9 \le k \le 13$ along with the lower bound r_k from (1).

8. Proving the existence of many frozen variables. Recall that in the view of coarsening as hypergraph stripping, we start by throwing $(k-1)m$ blue balls and m red balls uniformly and independently into n bins. Then, if a bin exists that contains blue balls only, we remove a random ball from that bin, another $k-2$ random blue balls, and a random red ball. If no such bin exists, we stop. We consider this a single step (the removal of precisely one unfrozen clause), and in the previous section we proved the following theorem.

THEOREM 16. *For every $k \ge 9$, there exists $c_k < r_k$ such that for all $r \ge c_k$, there exists $\alpha = \alpha(k,r) < 1$ such that w.h.p. every satisfying truth assignment reaches its coarsening fixed point within αm steps.*

To prove Theorem 16 we actually considered the process in which in each step we remove only one random blue ball (rather than $k-1$ random blue balls) and in which we continue to remove one random red ball per step (even when every nonempty bin contains at least one red ball). We proved that within its first αm steps this modified process will go through a configuration in which every nonempty bin contains at least one red ball, implying Theorem 16. Consider now the further modification in which the red balls carry numbers $1, \dots, m$ and where after the initial random placement of all balls we remove the lowest numbered red ball in each step (all $(k-1)m$ blue balls stay intact). It is trivial to see that the true coarsening process and this modified process can be coupled so that the configuration of red balls in the coarsening process is identical to their configuration in the red-ball-removal-only process up to the point where the coarsening process reaches its fixed point. Moreover, after that point, the set of red balls is decreasing in the red-ball-removal-only process and, therefore, the number of bins in this process with at least one red ball after αm steps provides a lower bound on the number of frozen variables in the coarsening fixed point. We will prove the following theorem.

THEOREM 17. *For k, r, and α as in Theorem 16, w.h.p. every satisfying assignment after αm steps of the red-ball-removal-only process is in a configuration where at least $(1-2/k)\cdot n$ bins contain at least one red ball.*

Theorems 16 and 17 combined readily imply that w.h.p. within αm steps every satisfying assignment has reached its coarsening fixed point and that fixed point has at least $(1-2/k)\cdot n$ bins with at least one red ball, i.e., Theorem 9.

Proof of Theorem 17. Recall the definitions of random variables b, m, q from (7)–(9) and of their associated large deviations parameterization from (13)–(15). Also, recall the parameterization $r = \lambda(2^k-1)/k$, the notation $\rho = \lambda(1+\delta)(1-\alpha)$, and the definition of s from (18).

To prove our claim it suffices to prove that for every $\delta, \epsilon, \zeta \ge -1$, either $\Omega > s$, i.e., the expected number of satisfying assignment with such an initial configuration

is $o(1)$, or that

$$f \equiv (1+\zeta)e^{-\rho} < 2/k, \tag{31}$$

i.e., the number of bins with no red ball after αm steps is smaller than $(2/k) \cdot n$.

Recall that our proof of the existence of frozen clauses amounts to showing that inequality (32) impies inequality (33), where

$$\Omega \leq s, \tag{32}$$

$$(1+\epsilon)(k-1)f < \alpha. \tag{33}$$

We will prove that (32) and (33) imply (31).

Recalling the definition of Ω from (17) we see that (32) implies

$$g \equiv \lambda\left(\omega(\delta) + (1+\delta)(k-1)f\omega(\epsilon)\right) \leq s \tag{34}$$

and, in particular, that

$$\min_{\delta} g = \lambda(1 - e^{-(k-1)f\omega(\epsilon)}) \leq s. \tag{35}$$

Define $x \equiv (k-1)f$ and assume that $x \geq 1$, for otherwise $f < 1/(k-1)$ and we are done as $1/(k-1) \leq 2/k$ for $k \geq 2$. Inequality (33) implies

$$\epsilon < \frac{1}{x} - 1, \tag{36}$$

and since $x \geq 1$, we see that $\epsilon < 0$. Since ω is decreasing in $(-1, 0)$ the fact $\epsilon < 0$ and (36) imply $\omega(\epsilon) > \omega(1/x - 1)$. Combined with (35) this last fact implies $1 - e^{-x\omega(1/x-1)} \leq s/\lambda$, which is equivalent to

$$1 - xe^{1-x} \leq \frac{s}{\lambda}.$$

Writing $r = c2^k \ln 2$ we see that $s/\lambda \leq \frac{1-c}{ck}$, and since $k \geq 9$ and $c \geq 4/5$ we get $s/\lambda < 1/36$ implying $x < 4/3$ and, thus, $f < 2/k$. □

REFERENCES

[1] D. Achlioptas, A. Coja-Oghlan, and F. Ricci-Tersenghi, *Random Formulas Have Exponentially Many Solution Clusters*, submitted.

[2] D. Achlioptas and C. Moore, *Random k-SAT: Two moments suffice to cross a sharp threshold*, SIAM J. Comput., 36 (2006), pp. 740–762.

[3] D. Achlioptas and A. Naor, *The two possible values of the chromatic number of a random graph*, Ann. of Math., 162 (2005), pp. 1335–1351.

[4] D. Achlioptas and C. Moore, *The chromatic number of random regular graphs*, in Proceedings of the 7th International Workshop on Randomization and Approximation Techniques in Computer Science, Lecture Notes in Comput. Sci. 3122, Springer, Berlin, 2004, pp. 219–228.

[5] D. Achlioptas, A. Naor, and Y. Peres, *The fraction of satisfiable clauses in a typical formula*, J. ACM, 54 (2), Article 10, (2007).

[6] D. Achlioptas, A. Naor, and Y. Peres, *Rigorous location of phase transitions in hard optimization problems*, Nature, 435 (2005), pp. 759–764.

[7] D. Achlioptas and Y. Peres, *The threshold for random k-SAT is* $2^k \ln 2 - O(k)$, J Amer. Math. Soc., 17 (2004), pp. 947–973.

[8] S. Aji and R. J. McEliece, *The generalized distributive law*, IEEE Trans. Inform. Theory, 46 (2000), pp. 325–343.

[9] M. Alekhnovich and E. Ben-Sasson, *Linear upper bounds for random walk on small density random 3CNFs*, SIAM J. Comput., 36 (2007), pp. 1248–1263.
[10] A. Braunstein and R. Zecchina, *Survey propagation as local equilibrium equations*, J. Stat. Mech., (2004), P06007.
[11] M.-T. Chao and J. Franco, *Probabilistic analysis of two heuristics for the 3-satisfiability problem*, SIAM J. Comput., 15 (1986), pp. 1106–1118.
[12] V. Chvátal and B. Reed, *Mick gets some (the odds are on his side)*, in Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science, IEEE Comput. Soc. Press, Los Alamitos, CA, 1992, pp. 620–627.
[13] O. Dubois and Y. Boufkhad, *A general upper bound for the satisfiability threshold of random r-SAT formulae*, J. Algorithms, 24 (1997), pp. 395–420.
[14] O. Dubois, Y. Boufkhad, and J. Mandler, *Typical random 3-SAT formulae and the satisfiablity threshold*, Electronic Colloquium on Computational Complexity, 10 (2003).
[15] A. M. Frieze and S. Suen, *Analysis of two simple heuristics on a random instance of k-SAT*, J. Algorithms, 20 (1996), pp. 312–355.
[16] M. Hajiaghayi and G. Sorkin, *The Satisfiability Threshold of Random 3-SAT is at least.* 3.52, IBM Technical Report: RC22942, 2003.
[17] A. Kamath, R. Motwani, K. Palem, and P. Spirakis, *Tail bounds for occupancy and the satisfiability threshold conjecture*, Random Structures Algorithms, 7 (1995), pp. 59–80.
[18] A. Kaporis, L. M. Kirousis, and E. G. Lalas, *The probabilistic analysis of a greedy satisfiability algorithm*, in Proceedings of the 10th Annual European Symposium on Algorithms, Lecture Notes in Comput. Sci. 2461, Springer, Berlin, 2002, pp. 574–585.
[19] A. Kaporis, L. M. Kirousis, and E. G. Lalas, *Selecting complementary pairs of literals*, Electron. Notes Discrete Math., 16 (2003), pp. 47–70.
[20] L. M. Kirousis, E. Kranakis, D. Krizanc, and Y. Stamatiou, *Approximating the unsatisfiability threshold of random formulas*, Random Structures Algorithms, 12 (1998), pp. 253–269.
[21] F. Krzaķala, A. Montanari, F. Ricci-Tersenghi, G. Semerjian, and L. Zdeborová, *Gibbs states and the set of solutions of random constraint satisfaction problems*, Proc. Natl. Acad. Sci. USA, 104 (2007), pp. 10318–10323.
[22] E. Maneva, E. Mossel, and M. J. Wainwright, *A new look at survey propagation and its generalizations*, J. ACM, 54 (4), Article 17, (2007).
[23] M. Mézard, G. Parisi, and R. Zecchina, *Analytic and algorithmic solution of random satisfiability problems*, Science, 297 (2002), pp. 812–815.
[24] N. C. Wormald, *Differential equations for random processes and random graphs*, Ann. Appl. Probab., 5 (1995), pp. 1217–1235.