EPJ manuscript No.
(will be inserted by the editor)

Solution Clustering in Random Satisfiability

Dimitris Achlioptas [a]

Department of Computer Science, University of California Santa Cruz

Received: date / Revised version: date

Abstract. For a large number of random constraint satisfaction problems, such as random k-SAT and random graph and hypergraph coloring, we have very good estimates of the largest constraint density for which solutions exist. All known polynomial-time algorithms for these problems, though, already fail to find solutions at much lower densities. To understand the origin of this gap we study how the structure of the space of solutions evolves in such problems as constraints are added. In particular, we show that for $k \geq 8$, 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. Moreover, inside each cluster most variables are frozen, i.e., take only one value.

PACS. 02.50.-r Probability theory, stochastic processes, and statistics – 75.10.Nr Spin-glass and other random models

1 Introduction

For a number of random Constraint Satisfaction Problems (CSP), we have very good rigorous estimates for the largest constraint density (ratio of constraints to variables) for which typical instances have solutions. For example [2], a random graph of average degree d is with high probability [1] k-colorable 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 $d < 2k \ln k$.

It is easy to color random graphs using a palette of $2k_d$ colors. The algorithm "'repeatedly pick a random vertex and assign it a random color not assigned to any of its neighbors" will w.h.p. succeed in coloring a random graph of average degree d if originally each vertex has $2k_d$ available colors. Equivalently, 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 random k-SAT the question is whether 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 *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 $r \geq 2^k \ln 2$. Recently, it was shown that random k-CNF formulas have satisfying assignments for densities very close to this upper bound [4]. Specifically, it was proven that for all $k \geq 3$,

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

As for the k-coloring problem, the lower bound in (1) is derived using the second moment method and, thus, yields no efficient algorithm for finding satisfying assignments. Indeed, the gap relative to algorithms for k-SAT is even greater than that for graph coloring: no polynomial algorithm is known that finds satisfying assignments of 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 [10], while for $k > 3$ from [9,15]. The best algorithmic lower bound for $k = 3$ is from [14], while for $k > 3$ it is from [12].

Similar results (and gaps relative to algorithms) also exist for random NAE k-SAT and hypergraph 2-coloring, regular random graph coloring, random Max k-SAT, and others (for example, see [3]). 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.

[a] This article is largely based on joint work with Federico Ricci-Tersenghi reported in [5].

[1] We will say that a sequence of events $\mathcal{E}_n$ occurs with high probability (w.h.p.) if $\lim_{n\to\infty} \Pr[\mathcal{E}_n] = 1$.

k	3	4	7	10	20	21
Best known upper bound for r_k^*	4.508	10.23	87.88	708.94	$726,817$	$1,453,635$
Best known lower bound for r_k	3.52	7.91	84.82	704.94	$726,809$	$1,453,626$
Best known algorithmic lower bound	3.52	5.54	33.23	172.65	$95,263$	$181,453$

Sparse random CSP have also been systematically studied by physicists in the past decades. Motivated by ideas developed in the study of materials, they 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 [19] developed an extremely efficient algorithm, called Survey Propagation (SP), for finding satisfying assignments of random formulas in the satisfiable regime.

Here we report on progress towards mathematical proof of some of the physically-motivated ideas underlying SP. In particular, we outline a proof establishing that for $k \geq 8$, significantly below the satisfiability threshold, the set of satisfying assignments fragments into exponentially many connected components. These components are relatively small in size, far apart from one another, and inside each one the majority of variables are "frozen", i.e., take only one value. As the formula density approaches the satisfiability 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. These rigorous results are in perfect agreement with the picture put forward using physical arguments.

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 the results. We emphasize that while both the discussion and the results are about random k-SAT, this is not strictly necessary: the 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 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 $v_i$ has 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 always setting v_i to 1 iff $p_i \geq 1/2$, then it is enough that if a variable ever takes the same value x in *all* satisfying assignments, x is the majority value in its 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 rigorously 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, again 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 identify a variable whose marginal can be determined with some confidence. 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 increasing performance:

UNIT-CLAUSE [8]: select a random variable and assign it a random value.

3-CLAUSE MAJORITY [7]: select a random variable and assign it its majority value among the 3-clauses.

SHORT-CLAUSE[12]: select a random shortest clause c, a random variable v in c, and set v so as to satisfy c.

HAPPIEST LITERAL [13]: satisfy a literal that appears in most clauses.

Each of the above heuristics computes marginals based on a different set of evidence, the content of which ranges from completely empty [8], to considering all the clauses containing each variable [13]. Correspondingly, the largest density for which these algorithms succeed on random 3-CNF formulas ranges from 8/3 for [8] to 3.42 for [13]. UNIT-CLAUSE, in fact, succeeds for every k as long as $r = O(2^k/k)$ and, as 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 (the bipartite graph where each constraint vertex is bound to the variables it binds.) One could hope that as d grows, such an algorithm would do well, perhaps even reach the satisfiability threshold. Unfortunately, the existence of connected components of satisfying assignments (clusters) with numerous frozen variables can induce long-range correlations among the variables, undermining such hopes.

To overcome the above issue, physicists hypothesized that the above clustering is the only significant source of long-range correlations. (Very) roughly speaking, this amounts to modeling 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. The results below imply that this simplified view of clusters is not very far off the truth for large k.

2 Statement of Results

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 non-empty set of clusters.*

Theorems 1 and 2 below build upon [4,17,18].

Theorem 1 *For every $k \geq 8$, there exists a value of $r < r_k$ and constants $\alpha_k < \beta_k < 1/2$ and $\epsilon_k > 0$ such that w.h.p. the set of satisfying assignments of $F_k(n, rn)$ consists of $2^{\epsilon_k n}$ cluster-regions, such that*

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

In other words, for all $k \geq 8$, at some point below the satisfiability threshold, the set of satisfying assignments decomposes into an exponential number of well-separated cluster-regions. The picture suggested by Theorem 1 comes in sharper focus for large k. In particular, for sufficiently large k, sufficiently close to the threshold, the cluster regions become arbitrarily small and maximally far apart, while remaining exponentially many. The following result gives a quantitative version of this fact.

Theorem 2 *For any $0 < \delta < 1/3$, if $r = (1-\delta)2^k \ln 2$, then for all $k \geq k_0(\delta)$, Theorem 1 holds with*

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

Remark 1 Theorems 1 and 2 remain valid for any definition of clusters in which a pair of assignments are deemed adjacent whenever their distance is at most $f(n) = o(n)$.

To "look inside" clusters and discuss the existence of frozen variables we need the following definition.

Definition 2 *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.*

Theorem 3 below asserts that for sufficiently large k, as we approach the satisfiability threshold, the fraction of frozen variables in every single cluster gets arbitrarily close to 1.

Theorem 3 *For every $\alpha > 0$ and all $k \geq k_0(\alpha)$, there exists $c_k^\alpha < r_k$, such that for all $r \geq c_k^\alpha$, w.h.p.* **every** *cluster of $F_k(n, rn)$ has at least $(1-\alpha)n$ frozen variables. As k grows,*

$$\frac{c_k^\alpha}{2^k \ln 2} \to \frac{1}{1+\alpha(1-\alpha)} \ .$$

By taking $\alpha = 1/2$ in Theorem 3 we see that for sufficiently large k, every cluster already has a majority of frozen variables at $r = (4/5 + \delta_k)2^k \ln 2$, with $\delta_k \to 0$, i.e., for a constant fraction of the satisfiable regime. More generally, Theorem 3 asserts that as k grows and the density approaches the threshold, clusters shrink in volume and grow further apart by having smaller and smaller internal entropy (more frozen variables).

The analysis that establishes Theorem 3 also implies

Corollary 1 *For every $k \geq 9$, there exists $r < r_k$ such that w.h.p.* **every** *cluster of $F_k(n, rn)$ has frozen variables.*

It remains open whether frozen variables exist for $k \leq 8$.

3 Clustering: Proof Sketch and Related Work

There are two main ingredients for proving Theorems 1 and 2. The first one excludes the possibility of pairs of truth assignments at certain Hamming distances and we discuss it in Section 3.1. The second one, discussed in Section 3.2, establishes the existence of a large total number of pairs of satisfying assignments and uses a simple packing argument to derive the existence of exponentially many clusters. In both cases we give the whole mathematical idea but leave out the (tedious) rigorous asymptotic analysis of the relevant functions, encapsulated as Theorem 4.

3.1 Forbidden distances and their implications for clustering

It is easy to show, see e.g., [1], that the expected number of pairs of satisfying assignments in $F_k(n, rn)$ with Hamming distance z is at most $\Lambda(z/n, k, r)^n$, where

$$\Lambda(\alpha, k, r) = \frac{2(1 - 2^{1-k} + 2^{-k}(1-\alpha)^k)^r}{\alpha^\alpha (1-\alpha)^{1-\alpha}} \ .$$

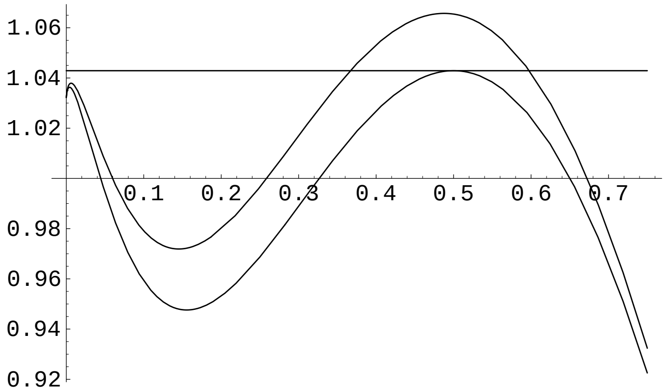

Fig. 1. Upper curve $\Lambda(\alpha, 8, 169)$ and lower curve $\Lambda_b(\alpha, 8, 169)$ for $\alpha \in [0, 3/4]$.

Thus, if for some k, r and $z = \alpha n$ we have $\Lambda(\alpha, k, r) < 1$, it immediately follows that w.h.p. in $F_k(n, rn)$ no pair of satisfying assignments has distance z. This observation was first made and used in [17]. In Figure 1 we draw the function Λ (upper curve), and a related function Λ_b (lower curve, to be discussed shortly), for $\alpha \in [0, 3/4]$ with $k = 8$ and $r = 169$. Recall that, by the results in [4], $F_8(n, 169n)$ is w.h.p. satisfiable and, thus, excluding the possibility of satisfying pairs at certain distances is a non-vacuous statement. We see that $\Lambda(\alpha, 8, 169) < 1$ for $\alpha \in [0.06, 0.26] \cup [0.68, 1]$, implying that w.h.p. in $F_8(n, 169n)$ there is no pair of satisfying assignments with Hamming distance αn for such values of α.

Establishing that there exists a distance z such that there are no pairs of assignments at distance z immediately implies an upper bound on the diameter of every cluster. This is because if a cluster C has diameter d, then it must contain pairs of solutions at every distance $1 \leq t \leq d$. To see this, take any pair $\sigma_1, \sigma_2 \in C$ that have distance d, any path from σ_1 to σ_2 in C, and observe that the sequence of distances from σ_1 along the vertices of the path must contain every integer in $\{1, \ldots, d\}$. Therefore, if $\Delta = \Delta_{k,r} \equiv \inf\{\alpha : \Lambda(\alpha, k, r) < 1\}$, then w.h.p. every cluster in $F_k(n, rn)$ has diameter at most Δn.

If we can further prove that $\Lambda(\alpha, k, r) < 1$ in an interval (α, β), then we can immediately partition the set of satisfying assignments into well-separated regions, as follows. Start with any satisfying assignment σ, let C be its cluster, and consider the set $R(C) \subseteq \{0, 1\}^n$ of truth assignments that have distance at most αn from C and the set $B(C) \subseteq \{0, 1\}^n$ of truth assignments that have distance at most βn from $R(C)$. Observe now that the set $B(C) \setminus R(C)$ cannot contain any satisfying truth assignments, as any such assignment would be at distance $\alpha n < d < \beta n$ from some assignment in C. Thus, the set of satisfying assignments in $R(C)$ is a union of clusters (cluster-region), all of which have distance at least βn from any cluster not in the region. Repeating this process until all satisfying assignments have been assigned to a cluster region gives exactly the subsets of Theorems 1 and 2. In fact, this arguments bounds the diameter of each entire cluster-region, not just of each cluster, by αn.

Remark 2 The argument above remains valid even if assignments are deemed adjacent whenever their distance is bounded by $f(n)$, for any $f(n) = o(n)$. As a result, Theorems 1 and 2 remain valid as stated for any definition of clusters in which assignments are deemed to belong in the same cluster if their distance is $o(n)$.

3.2 Establishing exponentially many clusters

Proving the existence of exponentially many non-empty cluster regions requires greater sophistication and leverages in a strong way the results of [4]. This is because having $\Lambda(\alpha, k, r) > 1$ for some α, k, r does *not* imply that pairs of satisfying assignments exist for such α, k, r: in principle, the behavior of Λ could be determined by a tiny minority of solution-rich formulas. Hence the need for the second moment method [1,4]. Specifically, say that a satisfying assignment σ is *balanced* if under σ the number of satisfied literal occurrences is in the range $km/2 \pm O(\sqrt{n})$, and let X be the number of balanced assignments in $F_k(n, rn)$. In [4], it was shown that $\mathbb{E}[X]^2 = \Lambda_b(1/2, k, r)^n$ and

$$\mathbb{E}[X^2] < C \times \max_{\alpha \in [0,1]} \Lambda_b(\alpha, k, r)^n \ ,$$

for some explicit function Λ_b and constant $C = C(k) > 0$. It was also shown that for all $r < 2^k \ln 2 - k$, the maximum of Λ_b occurs at $\alpha = 1/2$, implying that for such k, r we have $\mathbb{E}[X^2] < C \times \mathbb{E}[X]^2$. By the Payley-Zigmund inequality, this last fact implies that for any $t \leq \mathbb{E}[X]$,

$$\Pr[X > t] \geq \frac{(\mathbb{E}[X] - t)^2}{\mathbb{E}[X^2]} \ . \tag{2}$$

In [4], inequality (2) was applied with $t = 0$, establishing that for $r < 2^k \ln 2 - k$, $F_k(n, rn)$ has at least one (balanced) satisfying assignment with probability at least $1/C$. Taking $t = \mathbb{E}[X]/\text{poly}(n)$, implies that X is within a polynomial factor of its expectation $\Lambda_b(1/2, k, r)^{n/2}$, also with constant probability. Since the property "has more than q satisfying assignments" has a sharp threshold [11], this assertion implies that for every $r < 2^k \ln 2 - k$, w.h.p. $F_k(n, rn)$ has at least $\Lambda_b(1/2, k, r)^{n/2}/\text{poly}(n)$ satisfying assignments.

To prove that there are exponentially many clusters, it suffices to divide this lower bound for the total number of satisfying assignments with the following upper bound for the number of truth assignments in each cluster-region. Recall that $\Delta = \Delta_{k,r} \equiv \inf\{\alpha : \Lambda(\alpha, k, r) < 1\}$ and let

$$g(k, r) = \max_{\alpha \in [0, \Delta]} \Lambda(\alpha, k, r) \ .$$

If B is the expected number of pairs of truth assignments with distance at most Δn in $F_k(n, rn)$, it follows that $B < \text{poly}(n) \times g(k, r)^n$, since the expected number of pairs at each distance is at most $\Lambda(\alpha, k, r)^n$ and there are no more than $n + 1$ possible distances. By Markov's inequality, this implies that w.h.p. the number of pairs of truth assignments in $F_k(n, rn)$ that have distance at most Δn is $\text{poly}(n) \times g(k, r)^n$. Recall now that w.h.p. every cluster-region in $F_k(n, rn)$ has diameter at most Δn. Therefore, w.h.p. the total number of pairs of truth assignments in each cluster-region is at most $\text{poly}(n) \times g(k, r)^n$. Thus, if $g(k, r) < \Lambda_b(1/2, k, r)$, we can conclude that $F_k(n, rn)$ has at least

$$1/\text{poly}(n) \times \left(\frac{\Lambda_b(1/2, k, r)}{g(k, r)} \right)^{n/2}$$

cluster-regions. Indeed, the horizontal line in Figure 1 highlights that $g(8, 169) < \Lambda_b(1/2, 8, 169)$.

From the discussions in this section we see that to establish Theorems 1 and 2 it suffices to prove the following.

Theorem 4 *For every $k \geq 8$, there exists a value of $r < r_k$ and constants $\alpha_k < \beta_k < 1/2$ and $\epsilon_k > 0$ such that $\Lambda(\alpha, k, r) < 1$ for all $\alpha \in (\alpha_k, \beta_k)$ and*

$$\log_2 \left[\left(\frac{\Lambda_b(1/2, k, r)}{g(k, r)} \right)^{1/2} \right] > \epsilon_k \quad .$$

In particular, for any $0 < \delta < 1/3$ and all $k \geq k_0(\delta)$, if $r = (1 - \delta)2^k \ln 2$, this holds with

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

4 Frozen Variables: Survey Propagation and Related Work

For a cluster C, the string $\pi(C) = \pi_1(C), \pi_2(C), \ldots, \pi_n(C)$ denotes the **projection** of C and we will use the convention $\{0, 1\} \equiv *$, so that $\pi(C) \in \{0, 1, *\}^n$. Imagine now 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 $1 - x$ 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 of a cluster C, it is not at all clear how to compute $\pi(C)$ in time less than $|C|$. Survey Propagation (SP) can be interpreted as 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 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. The results below are closely related to this second assumption and establish that, indeed, the approximate cluster projections used in SP retain a significant amount of the information in the cluster projections. To make this last notion concrete and enhance intuition, we give below a self-contained, brisk discussion of Survey Propagation. For the sake of presentation this discussion is historically inaccurate. We attempt to restore history in Section 4.1.

As we said above, even if we are given a satisfying assignment σ, it is not obvious how to determine the projection of its cluster $C(\sigma)$. To get around this problem SP sacrifices information in the following manner.

Definition 3 *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 1 *For every formula F and $\sigma \in \mathcal{S}(F)$, 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 i-th coordinate, then the i-th 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 4 *The **core** of a cluster C is the unique coarsening fixed point of the truth assignments in C.*

By Lemma 1, 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 3 it suffices to prove that the core of every cluster has many non-$*$ variables.

Theorem 5 *For any $\alpha > 0$, let $k_0(\alpha)$ and c_k^α be as in Theorem 3. If $k \geq k_0$ and $r \geq c_k^\alpha$, then w.h.p. the coarsening fixed point of* **every** *$\sigma \in \mathcal{S}(F_k(n, rn))$ contains fewer than $\alpha \cdot n$ variables that take the value $*$.*

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 such a process will only assign $*$ to variables whose projection in $C(\sigma)$ is indeed $*$. Subsequent rounds, though, might not: a variable v will be 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 non-frozen* variables of each cluster makes this is 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.

Theorem 5 can be seen as evidence of the utility of this line of reasoning, since it shows that for sufficiently large densities, for all sufficiently large k, the coarsening fixed point of a satisfying assignment is *never* $(*,\ldots,*)$. Indeed, as we approach the satisfiability threshold from below, the fraction of frozen variables in the fixed point 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,

Definition 5 *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, by induction, that any string that results by applying coarsening steps to a satisfying assignment satisfies (i). 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, $(*,\cdots,*)$ is always a cover, even if F is unsatisfiable.

The Survey Propagation 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 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 [19] 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 4.1). Also, in [19], 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 non-frozen variables remain unset.

4.1 Related Work

Casting SP as an attempt to compute marginals over cores was done independently by Braunstein and Zecchina in [6] and Maneva, Mossel, and Wainwright in [16]. 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 applied to a factor graph encoding the set of all covers.

In [16], the authors give a number of formal correspondences between SP, Markov random fields and Gibbs sampling and note 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 non-essential 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, [16] gives experimental evidence that typical satisfying assignment of such formulas do not have non-trivial 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 "core-like" strings allowed under the proposed generalization.

5 Coarsening: Proof Sketch

Theorem 3 follows from Theorem 5 and Lemma 1. To prove Theorem 5, 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,

$$\mathbb{E}[X] = 2^n(1-2^{-k})^{rn} \cdot \Pr[\mathbf{0} \text{ is } \alpha\text{-coreless} \mid \mathbf{0} \text{ is satisfying}] \ .$$

Observe that conditioning on "**0** is satisfying" is exactly the same as "planting" the **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. Using the method of differential equations to analyze the typical behavior of the coarsening process, one can show that for every $k \geq 3$, there exists t_k^α such that

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

In particular, $t_k^1 \sim (2^k/k) \ln k$. We find it interesting that all algorithms that have been analyzed so far work only for densities below t_k^1, i.e., the point where the planted solution in the planted model has frozen variables.

To prove $\mathbb{E}[X] = o(1)$ for $r \gg t_k^\alpha$, one proves that $\Pr[\mathbf{0}$ is α-coreless $\mid$ $\mathbf{0}$ is satisfying$] < e^{-f(r)n}$ where f is such that for all $r \geq c_k^\alpha$,

$$2 \cdot \left(1 - \frac{1}{2^k}\right)^r \cdot e^{-f(r)} < 1 \ . \qquad (3)$$

Thus, for all such r we have $\mathbb{E}[X] = o(1)$ and Theorem 5 follows. Finally, to derive such a function f, it turns out that it is very helpful to model the coarsening process in the manner outlined below.

5.1 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. 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 $*$.

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 **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

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

It is convenient to work with 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 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 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 it passes the condition in Step 1, there are always suitable balls to remove. Thus, to get the function f, one gives a lower bound on the probability that this process exits within the first $i = \alpha m$ steps, for some carefully chosen $\alpha = \alpha(k, r) \in (0, 1)$.

References

1. D. Achlioptas and C. Moore, *The asymptotic order of the random k-SAT threshold*, in Proc. 43th Annual Symposium on Foundations of Computer Science (2002), 126–127.
2. D. Achlioptas and A. Naor, *The two possible values of the chromatic number of a random graph*, Annals of Mathematics, **162** (2005), 1333–1349.
3. D. Achlioptas, A. Naor, and Y. Peres, *Rigorous location of phase transitions in hard optimization problems*, Nature **435** (2005), 759–764.
4. D. Achlioptas and Y. Peres, *The threshold for random k-SAT is $2^k \ln 2 - O(k)$*, Journal of the American Mathematical Society **17** (2004), 947–973.
5. D. Achlioptas and F. Ricci-Tersenghi, *On the solution-space geometry of random constraint satisfaction problems*, in Proc. 38th Annual Symposium on Theory of Computing, 130–139.
6. A. Braunstein and R. Zecchina, *Survey propagation as local equilibrium equations*, Journal of Statistical Mechanics (2004), P06007.
7. M.-T. Chao and J. Franco, *Probabilistic analysis of two heuristics for the 3-satisfiability problem*, SIAM J. Comput. **15** (1986), 1106–1118.
8. V. Chvátal and B. Reed, *Mick gets some (the odds are on his side)*, in Proc. 33th Annual Symposium on Foundations of Computer Science (1992), 620–627.
9. O. Dubois and Y. Boufkhad, *A general upper bound for the satisfiability threshold of random r-SAT formulae*, Journal of Algorithms **24** (1997), 395–420.

10. O. Dubois, Y. Boufkhad, and J. Mandler, *Typical random 3-SAT formulae and the satisfiablity threshold*, Electronic Colloquium on Computational Complexity **10** (2003).
11. E. Friedgut, personal communication.
12. A. M. Frieze and S. Suen, *Analysis of two simple heuristics on a random instance of k-SAT*, Journal of Algorithms **20** (1996), 312–355.
13. A. Kaporis, L. M. Kirousis, and E. G. Lalas, *The probabilistic analysis of a greedy satisfiability algorithm*, in Proc. 10th Annual European Symposium on Algorithms, volume 2461 of *Lecture Notes in Computer Science*, Springer (2002), 574–585.
14. A. Kaporis, L. M. Kirousis, and E. G. Lalas, *Selecting complementary pairs of literals*, in Proc. LICS'03 Workshop on Typical Case Complexity and Phase Transitions, 2003.
15. L. M. Kirousis, E. Kranakis, D. Krizanc, and Y. Stamatiou, *Approximating the unsatisfiability threshold of random formulas*, Random Structures & Algorithms **12** (1998), 253–269.
16. E. Maneva, E. Mossel, and M. J. Wainwright, *A New look at Survey Propogation and its Generalizations*, in Proc. of SODA 2005, 1089–1098.
17. M. Mézard, T. Mora, and R. Zecchina, *Clustering of Solutions in the Random Satisfiability Problem*, Phys. Rev. Lett. **94** (2005), 197205. Also `arxiv:cond-mat/0504070`, April 4th 2005.
18. M. Mézard, T. Mora, and R. Zecchina, *Pairs of SAT Assignments and Clustering in Random Boolean Formulae*, `arxiv:cond-mat/0506053`, June 2nd 2005.
19. M. Mézard, G. Parisi, and R. Zecchina, *Analytic and Algorithmic Solution of Random Satisfiability Problems*, Science **297** (2002), 812–815.