Phase transition for Local Search on planted SAT

Andrei A. Bulatov   Evgeny S. Skvortsov
Simon Fraser University
{abulatov,evgenys}@cs.sfu.ca

Abstract

The Local Search algorithm (or Hill Climbing, or Iterative Improvement) is one of the simplest heuristics to solve the Satisfiability and Max-Satisfiability problems. It is a part of many satisfiability and max-satisfiability solvers, where it is used to find a good starting point for a more sophisticated heuristics, and to improve a candidate solution. In this paper we give an analysis of Local Search on random planted 3-CNF formulas. We show that if there is $\kappa < \frac{7}{6}$ such that the clause-to-variable ratio is less than $\kappa \ln n$ (n is the number of variables in a CNF) then Local Search whp does not find a satisfying assignment, and if there is $\kappa > \frac{7}{6}$ such that the clause-to-variable ratio is greater than $\kappa \ln n$ then the local search whp finds a satisfying assignment. As a byproduct we also show that for any constant ϱ there is γ such that Local Search applied to a random (not necessarily planted) 3-CNF with clause-to-variable ratio ϱ produces an assignment that satisfies at least γn clauses less than the maximal number of satisfiable clauses.

1 Introduction

A CNF formula over variables $x_1, \dots, x_n$ is a conjunction of clauses $c_1, \dots, c_m$ where each clause is a disjunction of one or more literals. A formula is said to be a k-CNF if every clause contains exactly k literals. In the problem k-SAT the question is, given a k-CNF, decide if it has a satisfying assignment (find such an assignment for the search problem). In the MAX-k-SAT problem the goal is to find an assignment that satisfies as many clauses as possible. The problem k-SAT for $k \geq 3$ is one of the first problems proved to be NP-complete problems and serves as a model problem for many algorithm and complexity concepts since then. In particular, Håstad [17] proved that the MAX-k-SAT problem is NP-hard to approximate within ratio better than 7/8. These worst case hardness results motivate the study of the typical case complexity of those problems, and a quest for probabilistic or heuristic algorithms with satisfactory performance, in the typical case. In this paper we analyze the performance of one of the simplest algorithms for (MAX-)k-SAT, the Local Search algorithm, on random planted instances.

The distribution. Let us start with planted instances. One of the most natural and well studied probability distributions on the set of 3-CNFs is the uniform distribution $\Phi(n, m(n))$ on the set of 3-CNFs with a given clauses-to-variables ratio [14]. It can be constructed and sampled as follows. Fix the number $m = m(n)$ of 3-clauses as a function of the number n of variables. The elements of $\Phi(n, m(n))$ are 3-CNFs generated by selecting $m = m(n)$ clauses over variables $x_1, \dots, x_n$. Clauses are chosen uniformly at random from the set of possible clauses, and so the probability of every 3-CNF from $\Phi(n, m(n))$ is the same. An important parameter of such CNFs is the *clause-to-variable ratio*, $\frac{m}{n}$, or *density* of the formula. We will use the density of a 3-CNF rather than the number of clauses, and so we write $\Phi(n, \varrho n)$ instead of $\Phi(n, m(n))$. Density can also be a function of n.

However, the typical case complexity for this distribution is not very interesting except for a very narrow range of densities. The reason is that the random 3-SAT under this distribution demonstrates a sharp satisfiability threshold in the density [2]. A random 3-CNF with density below the threshold (estimated to be around 4.2) is satisfiable whp (with high probability, meaning that the probability tends to 1 as n goes to infinity), and a 3-CNF with density above the threshold is unsatisfiable whp. Therefore the trivial algorithm outputting yes or no by just counting the density of a 3-CNF gives a right answer to 3-SAT whp. For more results on the threshold see [10, 11, 1, 18]. It is also known that, as the density grows, the number of clauses satisfied by a random assignment differs less and less from the maximal number of satisfiable clauses. If density is *infinite* (meaning it is an unbounded function of n), then whp this difference becomes negligible, i.e. $o(n)$. Therefore, distribution $\Phi(n, \varrho n)$ is not very interesting for MAX-3-SAT, at least when density is large, as one can get whp a very good approximation just by checking a random assignment.

A more interesting and useful distribution is obtained from $\Phi(n, \varrho n)$ by conditioning on satisfiability: such distribution is uniform and its elements are the satisfiable 3-CNFs. Then the problem is to find or approximate a satisfying assignment knowing it exists. Unfortunately, to date there are no techniques to tackle such problems (see, e.g., [6, 9]), particularly, to sample the satisfiable distribution. A good approximation for such a distribution is the planted distribution $\Phi^{\mathsf{plant}}(n, \varrho n)$, which is obtained from $\Phi(n, \varrho n)$ by conditioning on satisfiability by a specific "planted" assignment. To construct an element of a planted distribution we select an assignment of a set of n variables and then uniformly at random include ϱn clauses satisfied by the assignment selected. Some attempts have been made to define a better approximation of the satisfiable distribution, see, e.g. [20], however, the analysis of such distributions is difficult and it is not clear if they are closer to the distribution sought.

Another interesting feature of the planted distribution is that there is a hope that it is possible to design an algorithm that solves all planted instances whp. Some candidate algorithms were suggested in [6, 13, 21]. Algorithm from [13] and [21] use different approaches to solve planted 3-SAT of high density. Experiments show that the algorithm from [5] achieves the goal, but a rigorous analysis of this algorithm is not yet made. For a wider survey on SAT algorithms the reader is referred to [23, 7].

The algorithm. The Local Search algorithm (LS) is one of the oldest heuristics for SAT that has been around since the eighties. Numerous variations of this method have been proposed since then, see, e.g., [15, 25]. We study one of the most basic versions of LS, which, given a CNF, starts with a random assignment to its variables, and then on each step chooses at random a variable such that flipping this variable increases the number of satisfied clauses, or stops if such a variable does not exist. Thus LS finds a random local optimum accessible from the initial assignment.

LS has been studied before. The worst-case performance of pure LS is not very good: the only known lower bound for local optima of a k-CNF is $\frac{k}{k+1}m$ of clauses satisfied, where m is the number of all clauses [16]. In [19], it is shown that if density of 3-CNFs is linear, that is, $m = \Omega(n^2)$, then LS solves whp a random planted instance. Finally, in [8], we gave an estimation of the dependence of the number of clauses LS typically satisfies and the density of the formula.

Often visualization of the number of clauses satisfied by an assignment is useful: Assignments can be thought of as points of a landscape, and the elevation of a point corresponds to the number of clauses unsatisfied, the higher the point is, the less clauses it satisfies. It is suspected that 'topographic' properties of such a landscape are responsible for many complexity properties of satisfiability instances. For example, it is believed that the hardness of random CNFs whose density is close to the satisfiability threshold is due to the geometry of the satisfying assignments. They tend to concentrate around several centers, that make converging to a solution more difficult [7, 22]. As we shall see the performance of LS is closely related to geometric properties of the assignments, and so we hope that the study of LS may lead to a better understanding of those properties.

The behavior of other SAT/MAXSAT algorithms have been studied before. For example, the random walk has been analyzed in [24] and then in [3]. A message passing type algorithm, Warning Propagation, is studied in [12].

Our contribution. We classify the performance of LS for all densities higher than an arbitrary constant. In particular, we demonstrate that LS has a threshold in its performance. The main result is the following theorem.

Theorem 1 *(1) Let $\varrho \geq \kappa \cdot \ln n$, and $\kappa > \frac{7}{6}$. Then the local search whp finds a solution of an instance from $\Phi^{\mathsf{plant}}(n, \varrho n)$.*

(2) Let $c \leq \varrho \leq \kappa \cdot \ln n$, c a constant, and $\kappa < \frac{7}{6}$. Then the local search whp does not find a solution of an instance from $\Phi^{\mathsf{plant}}(n, \varrho n)$.

To prove part (1) of the theorem 1 we show that under those conditions all the local optima of a 3-CNF whp are either satisfying assignments, that is, global optima, or obtained by flipping almost all the values of planted solution, and so are located on the opposite side of the set of assignments. In the former case LS finds a satisfying assignment, while whp it does not reach the local optima of the second type. We also show that that for any constant density ϱ there is γ such that the assignment produced by LS on an instance from $\Phi^{\mathsf{plant}}(n, \varrho n)$ or $\Phi(n, \varrho n)$ satisfies at least γn clauses less than the maximal number of satisfiable clauses. Unfortunately, it is somewhat difficult to run computational experiments on CNFs of infinite density, as in order to have $\log n$ sufficiently large n must be prohibitively big. However, experiments we were able to conduct agree with the results.

Another region where LS can find a solution of the random planted 3-CNF is the case of very low density. Methods similar to Lemma 9 and Theorem 11 show that this low density transition happens around $\varrho \approx n^{-1/4}$. However, we do not go into details here.

Usually the main difficulty of analysis of algorithms for random SAT is to show that as an algorithm runs, some kind of randomness of the current assignment is kept. This property allows one to use 'card games', Wormald's theorem, and differential equations as in [1, 8], or relatively simple probabilistic constructions, such as martingales, as in [3]. For LS randomness cannot be assumed after just a few iterations of the algorithm, which makes its analysis more difficult. This is why the most difficult part of the proof is to identify to which extent assignments produced by LS as it runs remain random, while most of the probabilistic computations are fairly standard.

The paper is organized as follows. After giving several necessary definitions in Section 2, we prove in Section 3, that above the threshold established in Theorem 1 planted 3-CNFs do not have local optima that can be found by LS, other than satisfying assignments. In Section 4 we show that below the threshold there are many such optima, and that LS necessarily gets stuck into one of them.

2 Preliminaries

SAT. A 3-CNF is a conjunction of *3-clauses*. As we consider only 3-CNFs, we will always call them just clauses. Depending on the number of negated literals, we distinguish 4 types of clauses: $(-,-,-)$, $(+,-,-)$, $(+,+,-)$, and $(+,+,+)$. If φ is a 3-CNF over variables $x_1, \ldots, x_n$, an *assignment* of these variables is

INPUT: 3-SAT formula φ over variables $x_1, \ldots, x_n$.
OUTPUT: Boolean n-tuple $\vec{v}$, which is a local minimum of φ.
ALGORITHM:
choose uniformly at random a Boolean n-tuple $\vec{u}$
let U be the set of all variables x_i such that the number of clauses that can be made satisfied
  by flipping the value of x_i is strictly greater than the number of those made unsatisfied
while U is not empty
  pick uniformly at random a variable x_j from U
  change the value of x_j
  recompute U

Figure 1: Local Search

a Boolean n-tuple $\vec{u} = (u_1, \ldots, u_n)$, so the value of x_i is u_i. The *density* of a 3-CNF φ is the number $\frac{m}{n}$ where m is the number of clauses, and n is the number of variables in φ.

The *uniform* distribution of 3-CNFs of density ϱ (density may be a function of n), $\Phi(n, \varrho n)$ is the set of all 3-CNFs containing n variables and ϱn clauses equipped with the uniform probability distribution on this set. To sample a 3-CNF accordingly to $\Phi(n, \varrho n)$ one chooses uniformly and independently ϱn clauses out of $2^3 \binom{n}{3}$ possible clauses. Thus, we allow repetitions of clauses, but not repetitions of variables within a clause. *Random 3-SAT* is the problem of deciding the satisfiability of a 3-CNF randomly sampled accordingly to $\Phi(n, \varrho n)$. For short, we will call such a random formula a 3-CNF from $\Phi(n, \varrho n)$.

The *uniform planted* distribution of 3-CNF of density ϱ is constructed as follows. First, choose at random a Boolean n-tuple $\vec{u}$, a *planted* satisfying assignment. Then let $\Phi^{\mathsf{plant}}(n, \varrho n, \vec{u})$ be the uniform probability distribution over the set of all 3-CNFs over variables $x_1, \ldots, x_n$ with density ϱ and such that $\vec{u}$ is a satisfying assignment. For our goals we can always assume that $\vec{u}$ is the all-ones tuple, that is a 3-CNF belongs to $\Phi^{\mathsf{plant}}(n, \varrho n, \vec{u})$ if and only if it contains no clauses of the type $(-, -, -)$. We also simplify the notation $\Phi^{\mathsf{plant}}(n, \varrho n, \vec{u})$ by $\Phi^{\mathsf{plant}}(n, \varrho n)$. To sample a 3-CNF accordingly to $\Phi^{\mathsf{plant}}(n, \varrho n)$ one chooses uniformly and independently ϱn clauses out of $7 \binom{n}{3}$ possible clauses of types $(+, -, -), (+, +, -)$, and $(+, +, +)$. *Random Planted 3-SAT* is the problem of deciding the satisfiability of a 3-CNF from $\Phi^{\mathsf{plant}}(n, \varrho n)$.

The problems *Random MAX-3-SAT* and *Random Planted MAX-3-SAT* are the optimization versions of Random 3-SAT and Random Planted 3-SAT. The goal in these problems is to find an assignment that satisfies as many clauses as possible. Although the two problems usually are treated as maximization problems, it will be convenient for us to consider them as problems of minimizing the number of unsatisfied clauses. Since we always evaluate the absolute error of our algorithms, not the relative one, such transformation does not affect the results.

Local search. A formal description of the Local Search algorithm (LS) is given in Fig. 1. Observe that LS stops when reaches a local minimum of the number of unsatisfied clauses.

Given an assignment $\vec{u}$ and a clause c it will be convenient to say that c *votes* for a variable x_i to have value 1 if c contains literal x_i and its other two literals are unsatisfied. In other words if either (a) $\vec{u}$ assigns x_i to 0, c is not satisfied by $\vec{u}$, and it will be satisfied if the value of x_i is changed, or (b) the only literal in c satisfied by $\vec{u}$ is x_i. Similarly, we say that c votes for x_i if c contains the negation of x_i and its other two literals are not satisfied. Using this terminology we can define set U as the set of all variables such that the number of votes received to change the current value is greater than the number of those to keep it.

Random graphs. Probabilistic tools we use are fairly standard and can be found in the book [4].

Let φ be a 3-CNF with variables $x_1, \dots, x_n$. The *primal graph* $G(\varphi)$ of φ is the graph with vertex set $\{x_1, \dots, x_n\}$ and edge set $\{x_i x_j \mid \text{literals containing } x_i, x_j \text{ appear in the same clause}\}$. The *hypergraph* $H(\varphi)$ *associated with* φ is a hypergraph, whose vertices are the variables of φ and the edges are the 3-element sets of variables belonging to the same clause. Note that if $\varphi \in \Phi^{\mathsf{plant}}(n, \varrho n)$, then $H(\varphi)$ is a random 3-hypergraph with n vertices and ϱn edges, but $G(n)$ is not a random graph.

We will need the following properties that a graph $G(\varphi)$ of not too high density has.

Lemma 2 *Let $\varrho < \kappa \ln n$ for a certain constant κ, and let $\varphi \in \Phi^{\mathsf{plant}}(n, \varrho n)$.*

(1) For any $\alpha < 1$, whp all the subgraphs of $G(\varphi)$ induced by at most $O(n^\alpha)$ vertices have the average degree less than 5.

(2) The probability that $G(\varphi)$ has a vertex of degree greater than $\ln^2 n$ is $o(n^{-3})$.

Proof: (1) This part of the lemma is very similar to Proposition 13 from [12], and is proved in a similar way. Let S be a fixed set of variables with $|U| = \ell$. The number of 3-element sets of variables that include 2 variables from U is bounded from above by

$$\binom{\ell}{2}(n-2) \le \frac{1}{2}\ell^2 n.$$

For each of them the probability that this set is the set of variables of one of the random clauses chosen for φ (we ignore the type of the clause) equals

$$\frac{\kappa n \ln n}{\binom{n}{3}} = \frac{6\kappa \ln n}{(n-1)(n-2)}.$$

Thus, the probability that 2ℓ of them are included as clauses is at most

$$\binom{\frac{1}{2}\ell^2 n}{2\ell}\left(\frac{6\kappa \ln n}{(n-1)(n-2)}\right) \le \left(3e\kappa \cdot \frac{\ell \ln n}{n}\right)^{2\ell}.$$

Let $d = e(3e\kappa)^2$. Using the union bound, the probability that there exists a required set U with at most n^α variables is at most

$$\begin{aligned}
&\sum_{\ell=2}^{n^\alpha}\binom{n}{k}\left(\sqrt{\frac{d}{e}}\frac{\ell \ln n}{n}\right)^{2\ell} \\
&\quad\le \sum_{\ell=2}^{n^\alpha}\left(\frac{ne}{\ell}\cdot\frac{d}{e}\cdot\frac{\ell^2 \ln^2 n}{n^2}\right)^{\ell} \\
&\quad\le \sum_{\ell=2}^{n^\alpha}\left(d\frac{n^\alpha \ln^2 n}{n}\right)^{\ell} \\
&\quad= (dn^{\alpha-1}\ln^2 n)^2 \frac{1-(dn^{\alpha-1}\ln n)^{\ell-1}}{1-dn^{\alpha-1}\ln n} \\
&\quad= O(n^{2\alpha-2}\ln^4 n).
\end{aligned}$$

(2) The probability that the degree of a fixed vertex is at least $\ln^2 n$ is bounded from above by

$$\left(\frac{1}{n}\right)^{\ln^2 n}\binom{3\kappa n\ln n}{\ln^2 n}\le n^{-\ln^2 n}\left(\frac{3e\kappa n\ln n}{\ln^2 n}\right)^{\ln^2 n}=\left(\frac{3e\kappa}{\ln n}\right)^{\ln^2 n},$$

where $n^{-\ln^2 n}$ is the probability that some particular $\ln^2 n$ random clauses include x, and $\binom{3\kappa n\ln n}{\ln^2 n}$ is the number of $\ln^2 n$-element sets of clauses. Then it is not hard to see that

$$n\left(\frac{3e\kappa}{\ln n}\right)^{\ln^2 n}\longrightarrow 0,$$

as n goes to infinity. $\square$

Several times we need the following corollary from Azuma's inequality for supermartingales (see Lemma 1 from [26]).

Observation 3 *(1) Let Y_t be a supermartingale such that $\mathbf{E}(Y_{t+1}|Y_t)\le Y_t$ and $|Y_{t+1}-Y_t|<c$ for some c. Then $\mathbf{P}(Y_t-Y_0\ge bc)\le e^{-\frac{b^2}{2t}}$, for any $b>0$.*

(2) This inequality implies that if $\mathbf{E}(Y_{t+1}|Y_t)<Y_t-d$ and $|Y_{t+1}-Y_t|<c\le 1$ then the process $Z_t=Y_t-dt$ is a supermartingale and we have the following inequality

$$\mathbf{P}(Y_t-Y_0\ge bc)=\mathbf{P}\left(Z_t-Z_0\le\left(b+\frac{dt}{c}\right)\right)\le e^{-\frac{(b+dt)^2}{2tc^2}}\le e^{-bd}. \tag{1}$$

The following lemma is a simple corollary of Chernoff bound.

Lemma 4 *Let r,s be integers, $\theta<1$ a positive real, and let $\alpha_1,\dots,\alpha_r,\beta_1,\dots,\beta_s$ be some real constants. There are constants λ and C such that we have*

$$\mathbf{P}(X>Y)<Ce^{-\lambda\mathbf{E}(Y)} \tag{2}$$

for any random variables X and Y such that $\mathbf{E}(X)<\theta\mathbf{E}(Y)$ and $X=\sum_{i=0}^{r}\alpha_iX_i, Y=\sum_{i=0}^{s}\beta_iY_i$ for some binomial random variables $X_1,\dots,X_r,Y_1,\dots,Y_s$.

Proof: Let $\xi=\frac{1-\theta}{(r+s)\max(\max(\alpha_i),\max(\beta_i))}$. It is easy to see that event $X>Y$ implies occurrence of at least one of the events from the set

$$\mathcal{S}=\{\{X_i\ge\mathbf{E}(X_i)+\xi\mathbf{E}(Y)\}_{i\in\{0,\dots,r\}},\{Y_i\le\mathbf{E}(Y_i)-\xi\mathbf{E}(Y)\}_{i\in\{0,\dots,s\}}\}.$$

Indeed, inequality $X<Y$ can be derived from inequalities, opposite to the ones in $\mathcal{S}$ and $\mathbf{E}(X)<\theta\mathbf{E}(Y)$.

Application of Chernoff bound gives us inequalities

$$\begin{aligned}\mathbf{P}(|X-\mathbf{E}(X_i)|>\xi\mathbf{E}(Y)) &< e^{-\mathbf{E}(X_i)\xi^2\left(\frac{\mathbf{E}(Y)}{\mathbf{E}(X_i)}\right)^2/3}\le e^{-\xi^2\mathbf{E}(Y)\theta^{-2}/3},\\ \mathbf{P}(|Y-\mathbf{E}(Y_i)|>\xi\mathbf{E}(Y)) &< e^{-\mathbf{E}(Y_i)\xi^2\left(\frac{\mathbf{E}(Y)}{\mathbf{E}(Y_i)}\right)^2/3}\le e^{-\xi^2\mathbf{E}(Y)/3}.\end{aligned}$$

Thus if we set $\lambda=\xi^2/3$, $C=r+s$ then using union bound we can conclude that inequality (2) holds. $\square$

3 Success of Local Search

In this section we prove the first statement of the Theorem 1(1). This will be done as follows. First, we show that if a 3-CNF has high density, that is, greater than $\kappa \log n$ for some $\kappa > \frac{7}{6}$ then whp all the local minima that do not satisfy the CNF — we call such minima *proper* — concentrate very far from the planted assignment. This is the statement of Proposition 8 below. Then we use Lemma 5 to prove that starting from a random assignment LS whp does not go to that remote region. Therefore the algorithm does not get stuck to a local minimum that is not a solution.

Several times we will need the following observation that can be checked using the inequality $\binom{n}{\ell} \le \left(\frac{ne}{\ell}\right)^{\ell}$. For any n, γ, and α with $0 < \alpha < 1$

$$\binom{n}{\gamma n^{\alpha}} \le e^{(1-\alpha)\gamma n^{\alpha}\ln n - \gamma n^{\alpha}\ln\gamma + \gamma n^{\alpha}}. \tag{3}$$

We need the following two lemmas. Recall that the planted solution is the all-ones one.

Lemma 5 *Let $\varrho \ge \kappa \ln n$ for some constant κ, and let constants q_0, q_1 be such that $q_0 < q_1$. Whp any assignment with $q_0 n$ zeros satisfies more clauses than any assignment with $q_1 n$ zeros.*

Proof: Let $\vec{u}, \vec{v}$ be some vectors with $q_0 n$ and $q_1 n$ zeros, respectively. Let c be a random clause, then (1) with probability $\frac{1}{7}$ all its literals are positive, (2) with probability $\frac{3}{7}$ two literals are positive and similar (3) with probability $\frac{3}{7}$ one literal is positive. The probabilities that the clause is satisfied by $\vec{u}$ in these cases are $(1-q_0)^3$, $(1-q_0)^2 q_0$ and $(1-q_0)q_0^2$, respectively. Hence the total probability of a clause to be satisfied by $\vec{u}$ equals $\frac{(1-q_0)^3 + 3(1-q_0)^2 q_0 + 3(1-q_0)q_0^2}{7} = \frac{1-q_0^3}{7}$. A similar result holds for $\vec{v}$. Thus the expectation of the number of clauses satisfied by $\vec{u}$ and $\vec{v}$ in a random formula equals $\frac{1-q_0^3}{7}\kappa n \ln n$ and $\frac{1-q_1^3}{7}\kappa n \ln n$ respectively, thus applying lemma 4 we conclude that

$$\mathbf{P}\left(\ \vec{u} \text{ satisfies less than } \tfrac{1-(q_0^3+q_1^3)/2}{7}\kappa n \ln n \text{ clauses}\right) < e^{-\lambda' n \ln n},$$

for some $\lambda' > 0$. There are 2^n assignments, hence, application of the union bound finishes proof of the lemma. □

Lemma 6 *Let $\varrho \ge \kappa \ln n$ for some κ (not necessarily $> \frac{7}{6}$). There is $\alpha < 1$ such that for $\varphi \in \Phi^{\mathsf{plant}}(n, \varrho n)$ whp for any proper local minimum $\vec{u}$ of φ the number of variables assigned to 0 by $\vec{u}$ is either less than n^{α}, or greater than $\frac{9n}{10}$.*

Proof: Let $M, |M| = \ell$ be the set of all variables that $\vec{u}$ assigns to 0. Let $\mathcal{B}_M^{each}$ be event "for every $x_i \in M$ the number of clauses voting for x_i to be 1 is less than or equal to the number of clauses voting for x_i to be 0". Since $\vec{u}$ is a local minimum, $\mathcal{B}_M^{each}$ is the case for $\vec{u}$. It is easy to see that event $\mathcal{B}_M^{each}$ implies event $\mathcal{B}_M^{all}$ = "the total number of votes given by clauses for variables in M to be 1 is less than or equal to the total number of votes given by clauses for variables in M to be 0". To bound the probability of $\mathcal{B}_M^{each}$ we will bound the probability of $\mathcal{B}_M^{all}$.

Let c be a random clause. It can contribute from 0 to 3 votes for variables in M to be one and 0 or 1 vote for them to remain zero. Let us compute, for example, the probability that it contributes exactly two votes for variables in M to become one. It happens if c is of type $(+,+,-)$, both its positive variables are in M and the negative variable is outside of M. Probability of this event is $\frac{3}{7}\ell^2 n^{-2}(1-\ell/n)$. So the expectation of the number of clauses voting for exactly 2 variables in M to be 1 is $\frac{3}{7}\ell^2 n^{-1}(1-\ell/n)\kappa\ln n$. The expectations of the numbers of clauses voting for three and one variables to be 1 are $\frac{1}{7}\ell^3 n^{-2}\kappa\ln n$ and $\frac{3}{7}(1-\frac{\ell}{n})^2\ell\kappa\ln n$, respectively.

A clause votes for a variable in M to remain 0 if its type is $(+,-,-)$, one of its negative literals is not in M, and two other literals are in M, or if its type is $(+,+,-)$ and all the variables in it belong to M. Thus the expectation of the number of clauses voting for variables in M to remain 0 is $\frac{3}{7}\kappa\ln n\left(2\ell^2 n^{-1}(1-\ell/n)+\ell^3 n^{-2}\right)$.

Hence the expectation of the number of votes for variables in M to flip equals

$$\mathbf{E}\left(\text{votes for a flip}\right) = \kappa\ln n \times \left(3\cdot\frac{1}{7}\ell^3 n^{-2} + 2\cdot\frac{3}{7}\ell^2 n^{-1}(1-\ell/n) + 1\cdot\frac{3}{7}\ell(1-\ell/n)^2\right)$$

and expectation of the number of votes for variables in M to remain 0 equals

$$\mathbf{E}\left(\text{votes for status quo}\right) = \kappa\ln n \times \left(\frac{6}{7}\ell^2 n^{-1}(1-\ell/n) + \frac{3}{7}\ell^3 n^{-2}\right).$$

If $\ell < \frac{9}{10}n$ then

$$\begin{aligned}\frac{\mathbf{E}\left(\text{votes for status quo}\right)}{\mathbf{E}\left(\text{votes for a flip}\right)} &= \frac{6\ell(n-\ell)+3\ell^2}{6\ell(n-\ell)+3\ell^2+3(n-\ell)^2} = 1-\frac{3(n-\ell)^2}{6\ell(n-\ell)+3\ell^2+3(n-\ell)^2}\\ &< 1-\frac{3\cdot\frac{1}{100}n^2}{12n^2} = 1-\frac{1}{400}.\end{aligned}$$

Therefore we can apply Lemma 4 to the votes for and against 0s and get the following bound $\mathbf{P}\left(\mathcal{B}_M^{all}\right) < e^{-\lambda\mathbf{E}(\text{votes for a flip})}$ for some $\lambda > 0$. Then we can bound number of votes for a flip from below by $\delta\ell\ln n$ for some constant δ and we can bound the number of sets M of size ℓ as

$$\#(\text{M of size }\ell) = \binom{n}{\ell} \le \left(\frac{ne}{\ell}\right)^\ell = e^{\ell\ln(n/\ell)+\ell}.$$

Therefore if

$$\ell\ln(n/\ell)+\ell < \delta\ell\ln n$$

then union bound implies that whp there is no set M such that $\mathcal{B}_M^{all}$ happens. It is easy to see that for $\ell > n^\alpha$ and α that is close enough to 1 the above inequality holds, which finishes the proof of the lemma. □

Now suppose that $\vec{u}$ is a proper local minimum of $\varphi \in \Phi^{\mathsf{plant}}(n,\varrho n)$. There is a clause $c\in\varphi$ that is not satisfied by $\vec{u}$. Without loss of generality, let the variables in c be x_1, x_2, x_3, and let the variable assigned 0 be x_1. Thus, clause c votes for x_1 to be flipped to 1. Since $\vec{u}$ is a local minimum there must a clause that is satisfied, that becomes unsatisfied should x_1 flipped. We call such a clause a *support* clause for the 0 value of x_1. In any support clause the supported variable is negated, and therefore any support clause has the type $(+,-,-)$ or $(+,+,-)$. A variable of a CNF is called k*-isolated* if it appears positively in at most k clauses of the type $(+,-,-)$. The *distance* between variables of a CNF φ is the length of the shortest path in $G(\varphi)$ connecting them.

Lemma 7 *If $\kappa > \frac{7}{6}$ and $\varrho \geq \kappa \ln n$ then for any integers $d_1, d_2 \geq 1$ and for a random $\varphi \in \Phi^{\mathsf{plant}}(n, \varrho n)$ whp there are no two d_1-isolated variables within distance d_2 from each other.*

Proof: Let x be some variable. The probability that it is d_1-isolated can be computed as

$$\begin{aligned}
\mathbf{P}\left(x \text{ is } d_1\text{-isolated}\right) &= d_1 \cdot \binom{\kappa n \ln n}{d_1} \left(1 - \frac{3}{7n}\right)^{\kappa n \ln n - d_1} \left(\frac{3}{7n}\right)^{d_1} \\
&\leq d_1 (\kappa n \ln n)^{d_1} \left(1 - \frac{3}{7n}\right)^{\kappa n \ln n} \left(1 - \frac{3}{7n}\right)^{-d_1} \left(\frac{7}{3} n\right)^{-d_1} \\
&\sim d_1 \left(1 - \frac{3}{7n}\right)^{-d_1} \left(\frac{7\kappa}{3} \ln n\right)^{d_1} e^{-\frac{3}{7}\kappa \ln n} \\
&= O(n^{-\frac{3\kappa}{7} + \varepsilon}),
\end{aligned}$$

for any $\epsilon > 0$.

By Lemma 2(2), the degree of every vertex of $G(\varphi)$ whp does not exceed $\ln^2 n$. Hence, there are at most $\ln^{2d_2} n$ vertices at distance d_2 from x. Applying the union bound we can estimate the probability that there is a d_1-isolated vertex at distance d_2 from x as $O(\ln^{2d_2} n \cdot n^{-\frac{3}{7}\kappa})$. Finally, taking into account the probability that x itself is d_1-isolated, and applying the union bound over all vertices of $G(\varphi)$ we obtain that the probability that two d_1-isolated vertices exists at distance d_2 from each other can be bounded from above by

$$n \cdot O(n^{-\frac{3\kappa}{7}}) \cdot O(\ln^{2d_2} n \cdot n^{-\frac{3}{7}\kappa}) = O(\ln^{2d_2} n \cdot n^{1-\frac{6}{7}\kappa}).$$

Thus for $\kappa > \frac{7}{6}$ whp there are no two such vertices. □

Proposition 8 *Let $\varrho \geq \kappa \cdot \ln n$, and $\kappa > \frac{7}{6}$. Then whp proper local minima of a 3-CNF from $\Phi^{\mathsf{plant}}(n, \varrho n)$ have at most $\frac{n}{10}$ ones.*

Proof: Let $\varphi \in \Phi^{\mathsf{plant}}(n, \varrho n)$ be a random planted instance. Suppose that $\vec{u}$ is a proper local minimum that has more than $\frac{n}{10}$ ones. We use the following observation. Let c be a clause not satisfied by $\vec{u}$. Then it contains at least one variable x_i that is assigned to zero by $\vec{u}$. The assignment $\vec{u}$ is a local minimum, so there must be a clause c' that is satisfied only by x_i. Hence, c' is a support clause, and contains a variable x_j which is assigned to zero by $\vec{u}$. Variables x_i and x_j are at distance 1. Setting $d_1 = 11$ and $d_2 = 1$, by Lemma 7, we conclude that one of them is not 11-isolated.

Set $d_1 = 11$, $d_2 = 3$ and consider the set Z of all variables assigned to zero by $\vec{u}$ that are not 11-isolated. By the observation above this set is non-empty. On the other hand, by Lemma 6, $|Z|$ is $O(n^{\alpha})$ for some $\alpha < 1$. Consider $x \in Z$. It appears positively in at least 10 clauses of the type $(+, -, -)$. Each of these clauses is either unsatisfied or contains a variable assigned to 0. Suppose there are k unsatisfied clauses among them. Since $\vec{u}$ is a local minimum, to prevent x from flipping, x must be supported by at least k support clauses, each of which contains a variable assigned to 0. Thus, at least 6 neighbors of x in $G(\varphi)$ are assigned to 0. Any two neighbors of x are at distance 2. By Lemma 7 at least 5 of the neighbors assigned to 0 are not 11-isolated, and therefore belong to Z. Thus the subgraph induced by Z in $G(\varphi)$ has the average degree greater than 5, which is not possible by Lemma 2(1). □

Now we are in a position to prove statement (1) of Theorem 1.

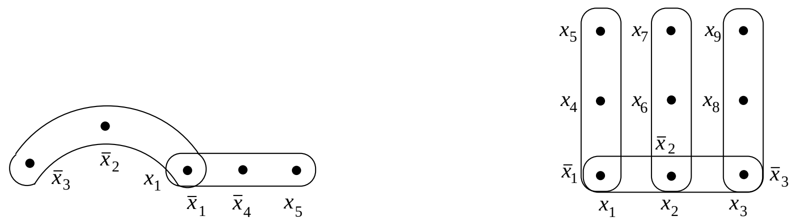

Figure 2: Caps and crowns

Proof: [of Theorem 1(1)] By Lemma 5 for a $\varphi \in \Phi^{\mathsf{plant}}(n, \varrho n)$ whp any assignment with dn variables equal to 1, where $\frac{1}{3} \leq d \leq \frac{2}{3}$, satisfies more clauses than any assignment with $\frac{n}{10}$ equal to 1. Then, whp a random initial assignment for LS assigns between $\frac{1}{3}$ and $\frac{2}{3}$ of all variables to 1. Therefore, whp LS never arrives to a proper local minimum with less than $\frac{n}{10}$ variables equal to 1, and, by Proposition 8, to any proper local minimum. $\square$

4 Failure of Local Search

We now prove statement (2) of Theorem 1. The overall strategy is the following. First, we show, Proposition 10, that in contrast to the previous case there are many proper local minima in the close proximity of the planted assignment. Then we show, Proposition 12, that those local minima are located so that they intercept almost every run of LS, and thus almost every run is unsuccessful.

We start off with a technical lemma. A pair of clauses $c_1 = (x_1, \overline{x}_2, \overline{x}_3)$, $c_2 = (\overline{x}_1, \overline{x}_4, x_5)$ is called a *cap* if x_1, x_5 are 1-isolated, that is they do not appear in any clause of the type $(+, -, -)$ except for c_1 and c_2, respectively, and x_2, x_3 are not 0-isolated (see Figure 2(a)). We denote equality $f(n) = g(n)(1 + o(n))$ by $f(n) \sim g(n)$.

Lemma 9 *Let $n^{-\frac{1}{4}} < \varrho \leq \kappa \cdot \ln n$, and $\kappa < \frac{7}{6}$. There is α, $0 < \alpha < 1$, such that whp a random planted CNF $\varphi \in \Phi^{\mathsf{plant}}(n, \varrho n)$ contains at least n^{α} caps.*

Proof: The proof is fairly standard, see, e.g. the proof of Theorem 4.4.4 in [4]. We use the second moment method. The result follows from the fact that a cap has properties similar to the properties of *strictly balanced graphs*, see [4]. Take some n, and let X be a random variable equal to the number of caps in a 3-CNF $\varphi \in \Phi^{\mathsf{plant}}(n, \varrho n)$. Straightforward calculation shows that the probability that a fixed 5-tuple of variables is a cap is $\sim \varrho^4 n^{-4-\frac{6}{7}\frac{\varrho}{\ln n}}$. Therefore $\mathbf{E}(X) \sim \varrho^4 n^{1-\frac{6}{7}\frac{\varrho}{\ln n}}$.

Let S be a fixed 5-tuple of variables, say, $S = (x_1, x_2, x_3, x_4, x_5)$, and A_S denote the event that S forms a cap. For any other 5-tuple T, the similar event is denoted by A_T, and we write $A_T \asymp A_S$ if these two events are not independent. By Corollary 4.3.5 of [4] it suffices to show that

$$\Delta^* = \sum_{T \asymp S} \mathbf{P}(A_T \mid A_S) = o(\mathbf{E}(X)).$$

Let $T = (y_1, y_2, y_3, y_4, y_5)$. It is not hard to see that the only cases when A_T and A_S are not independent and the probability $\mathbf{P}(A_T \mid A_S)$ is significantly different from 0 is: $y_1 = x_1$ and $\{y_2, y_3\} = \{x_2, x_3\}$, or $y_1 = x_5$ and $\{y_2, y_3\} = \{x_1, x_4\}$, or $y_5 = x_1$ and $\{y_1, y_4\} = \{x_2, x_3\}$, or $y_5 = x_5$ and $\{y_1, y_4\} = \{x_1, x_4\}$. Then, as before, it can be found that in each of these cases $\mathbf{P}(A_T \mid A_S) = O(\varrho^4 n^{-2-\frac{3}{7}\frac{\varrho}{\ln n}})$.

Finally,

$$\begin{aligned}\Delta^* &= \sum_{T \asymp S} \mathbf{P}\left(A_T \mid A_S\right) = n^2 \mathbf{P}\left(A_T \mid A_S\right) = n^2 \cdot O(\varrho^4 n^{-2-\frac{3}{7}\frac{\varrho}{\ln n}}) \\ &= O(\varrho^4 n^{-\frac{3}{7}\frac{\varrho}{\ln n}}) = o(\mathbf{E}\left(X\right)).\end{aligned}$$

We can choose $\alpha = 1 - \frac{6}{7}\kappa$ if $\varrho \geq 1$, and $\alpha = 1 - 4\nu$ if $1 > \varrho > n^{-\nu}$ for $\nu < \frac{1}{4}$. $\square$

Proposition 10 *Let $\varrho \leq \kappa \cdot \ln n$, and $\kappa < \frac{7}{6}$. Then there is α, $0 < \alpha \leq 1$, such that a 3-CNF from $\Phi^{\mathsf{plant}}(n, \varrho n)$ whp has at least n^α proper local minima.*

Proof: Let $c_1 = (x_1, \overline{x}_2, \overline{x}_3)$, $c_2 = (\overline{x}_1, \overline{x}_4, x_5)$ be a cap and $\vec{u}$ an assignment such that $u_3 = u_5 = 0$, and $u_i = 1$ for all other i. It is straightforward that $\vec{u}$ is a proper local minimum. By Lemma 9, there is α such that whp the number of such minima is at least n^α. $\square$

Before proving Proposition 12, we note that a construction similar to caps helps evaluate the approximation rate of the local search in the case of constant density on planted and also on arbitrary CNFs. A subformula $c = (x_1, x_2, x_3), c_1 = (\overline{x}_1, x_4, x_5), c_2 = (\overline{x}_2, x_6, x_7), c_3 = (\overline{x}_3, x_8, x_9)$ is called a *crown* if the variables $x_1, \ldots, x_9$ do not appear in any clauses other than c, c_1, c_2, c_3 (see Fig. 2(b)). The crown is satisfiable, but the all-zero assignment is a proper local minimum. For a CNF φ and an assignment $\vec{u}$ to its variables, by $\mathsf{OPT}(\varphi)$ and $\mathsf{sat}(\vec{u})$ we denote the maximal number of simultaneously satisfiable clauses and the number of clauses satisfied by $\vec{u}$, respectively.

Theorem 11 *If density ϱ is such that $n^{-\nu} \leq \varrho \leq \kappa \ln n$ for some $\nu < 1/4$ and $\kappa < 1/27$, then there is $\gamma_\varrho = \frac{1}{o(n)}$ such that whp Local Search on a 3-CNF $\varphi \in \Phi(n, \varrho n)$ ($\varphi \in \Phi^{\mathsf{plant}}(n, \varrho n)$) returns an assignment $\vec{u}$ such that $\mathsf{OPT}(\varphi) - \mathsf{sat}(\vec{u}) \geq \gamma(\varrho) \cdot n$, where $\mathsf{OPT}(\varphi)$ denotes the maximal number of clauses in φ that can be simultaneously satisfied and $\mathsf{sat}(\vec{u})$ denotes the number of clauses satisfied by $\vec{u}$.*

If ϱ is constant then γ_ϱ is also constant.

Proof: As in the proof of Lemma 9, it can be shown that for ϱ that satisfies conditions of this theorem there is $\gamma' = \frac{1}{o(n)}$ such that whp a random [random planted] formula has at least $\gamma' n$ crowns. If ϱ is a constant, γ' is also a constant. For a random assignment $\vec{u}$, whp the variables of at least $\frac{\gamma'}{1024} n$ crowns are assigned zeroes. Such an all-zero assignment of a crown cannot be changed by the local search. $\square$

Then we move on to proving Proposition 12.

Proposition 12 *Let $\varrho \leq \kappa \cdot \ln n$, and $\kappa < \frac{7}{6}$. The local search on a 3-CNF from $\Phi^{\mathsf{plant}}(n, \varrho n)$ whp ends up in a proper local minimum.*

If $\varrho = o(\ln n)$ then Proposition 12 follows from Theorem 11. So in what follows we assume that $\varrho > \kappa' \cdot \ln n$. The main tool of proving Proposition 12 is coupling of local search (LS) with the algorithm STRAIGHT DESCENT (SD) that on each step chooses at random a variable assigned to 0 and changes its value to 1. Obviously SD is not a practical algorithm, since to apply it we need to know the solution. For the purposes of our analysis we modify SD as follows. At each step SD chooses a variable at random, and if it is assigned 0 changes its value (see Fig. 4(a)). The algorithm LS is modified in a similar way (see Fig. 4(b)).

It is easy to see that the vector obtained by SD at step t does not depend on the formula. And since SD treats all variables equally we can make the following

INPUT: $\varphi \in \Phi^{\mathsf{plant}}(n, \varrho n)$ with the all-ones solution, Boolean tuple $\vec{u}$,
OUTPUT: The all-ones Boolean tuple.
ALGORITHM:
while there is a variable assigned 0
pick uniformly at random variable x_j from the set of all variables
if $u_j = 0$ **then set** $u_j = 1$

(a)

INPUT: 3-SAT formula φ, Boolean tuple $\vec{u}$,
OUTPUT: Boolean tuple $\vec{v}$, which is local minima of φ.
ALGORITHM:
while $\vec{u}$ is not a local minima
pick uniformly at random variable x_j from the set of all variables
if the number of clauses that can be made satisfied by flipping the value of x_i is strictly greater than the number of those made unsatisfied **then set** $u_j = \overline{u}_i$

(b)

Figure 3: Straight Descent (a) and Modified Local Search (b)

Lemma 13 *If SD starts its work at a random vector with m_0 ones and after step t, $t \le n - m_0$, it arrives to a vector with m ones, then this vector is selected uniformly at random from all vectors with m ones.*

Proof: Let us denote the probability that at step t SD arrives to vector $\vec{u}$, conditional to it starts from a vector with m_0 ones, by $\mathbf{P}\,(\vec{u}, t, m_0)$. We prove by induction on t that $\mathbf{P}\,(\vec{u}_1, t, m_0) = \mathbf{P}\,(\vec{u}_2, t, m_0)$ for any $\vec{u}_1, \vec{u}_2$ with m ones. We denote this number by $\mathbf{P}\,(t, m, m_0)$. As the starting vector is random, it is obvious for $t = 0$. Then for $t > 1$ and any vector $\vec{u}$ with m ones we have

$$\begin{aligned}\mathbf{P}\,(\vec{u}, t, m_0) &= \mathbf{P}\,(\vec{u}, t-1, m_0) \cdot \frac{m}{n} + \sum_{\vec{u}'} \mathbf{P}\left(\vec{u}', t-1, m_0\right) \cdot \frac{1}{n} \\ &= \mathbf{P}\,(t-1, m, m_0) \cdot \frac{m}{n} + \mathbf{P}\,(t-1, m-1, m_0) \cdot \frac{m}{n},\end{aligned}$$

where n is the number of variables in the formula and $\vec{u}'$ goes over all vectors that can be obtained from $\vec{u}$ by flipping a one into zero. It does not depend on a particular vector $\vec{u}$. □

We will frequently use the following two properties of the algorithm SD.

Lemma 14 *Whp the running time of SD does not exceed $2n \ln n$.*

Proof: For a variable x_i the probability that it is not considered for t steps equals $\left(1 - \frac{1}{n}\right)^t$. So for $t = 2n \ln n$ this probability equals $\left(1 - \frac{1}{n}\right)^{2n \ln n} \le e^{-2 \ln n} = n^{-2}$. Applying the union bound over all variables we obtain the required statement. □

Given 3-CNF φ and an assignment $\vec{u}$ we say that a variable x_i is k-*righteous* if the number of clauses voting for it to be one is greater by at least k than the number of clauses voting for it to be zero. Let $\varphi \in \Phi^{\mathsf{plant}}(n, \varrho n)$ and $\vec{u}$ be a Boolean tuple. The *ball* of radius m with the center at $\vec{u}$ is the set of all tuples of the same length as $\vec{u}$ at Hamming distance at most m from $\vec{u}$. Let $f(n)$ and $g(n)$ be arbitrary functions and d be an integer constant. We say that a set S of n-tuples is $(g(n), d)$-*safe*, if for any $\vec{u} \in S$ the number of variables that are not d-righteous does not exceed $g(n)$. A run of SD is said to be $(f(n), g(n), d)$*-safe* if at each step of this run the ball of radius $f(n)$ with the center at the current assignment is $(g(n), d)$-safe.

Lemma 15 *Let $\varrho > \kappa' \cdot \ln n$ for some κ'. For any constants γ and d there is a constant $\alpha_1 < 1$ such that, for any $\alpha > \alpha_1$, whp a run of SD on $\varphi \in \Phi^{\mathsf{plant}}(n, \varrho n)$ is $(\gamma n^\alpha, n^\alpha, d)$-safe.*

Proof: Consider a run of SD on $\varphi \in \Phi^{\mathsf{plant}}(n, \varrho n)$ with a random initial assignment. If SD starts its work at a tuple with m_0 ones, then at step t it has $m \le m_0 + t$ ones. Then by Lemma 13 if at step t the current assignment of SD has m ones then it is drawn uniformly at random from all vectors with m ones. Event *Unsafe* $=$ "run of SD is not $(\gamma n^\alpha, n^\alpha, d)$-safe" is a union of events "at step t of SD's run the ball of radius γn^α with the center at the current assignment is not (n^α, d)-safe". We will use the union bound to show that probability of *Unsafe* is small.

Let $\vec{u}$ be a Boolean n-tuple having pn positions filled with 1s. Since whp the number of 1s in the initial assignment is at least $\frac{n}{3}$, for every step the number of 1s is at least $\frac{n}{3}$. Let M be an arbitrary set of variables with $|M| = n^\alpha$. We consider events $\mathcal{B}_M^{each} =$ "every variable $x_i \in M$ is not k-righteous" and $\mathcal{B}_M^{all} =$ "the total number of votes given by clauses for variables in M to be 1 does not exceed the total number of votes given by clauses for variables in M to be 0 plus $|M| \cdot k$."

The same technique as in Lemma 6 can be used to show that the probability of $\mathcal{B}_M^{all}$ and consequently the probability of $\mathcal{B}_M^{each}$ is bounded above by $e^{-\lambda' n^\alpha \ln n}$ for some constant λ', not dependent on α. By inequality (3), there are at most $\gamma n^\alpha \cdot e^{\gamma(1-\alpha) n^\alpha \ln n \cdot (1+o(1))}$ distinct assignments in the γn^α-neighborhood of SD and $e^{n^\alpha (1-\alpha) \ln n (1+o(1))}$ distinct subsets of size n^α. So for α close to 1 the union bound implies that $\mathcal{B}_M^{each}$ whp does not take place for any tuple, any subset of variables at any step which completes the proof of the lemma. $\square$

For CNFs ψ_1, ψ_2 we denote by $\psi_1 \wedge \psi_2$ their conjunction.

We will need formulas that obtained from a random formula by adding some clauses in an 'adversarial' manner. Following [21] we call distributions for such formulas *semi-random*. However, the type of semi-random distributions we need is different from that in [21]. Let $\eta < 1$ be some constant. A formula φ is sampled according to semi-random distribution $\Phi_\eta^{\mathtt{plant}}(n, \varrho n)$ if $\varphi = \varphi' \wedge \psi$, where φ' is sampled according to $\Phi^{\mathsf{plant}}(n, \varrho n)$ and ψ contains at most n^η clauses and is given by an adversary.

Corollary 16 *If $\varphi' \in \Phi_\eta^{\mathtt{plant}}(n, \varrho n)$ then for any constants γ and d there is a constant $\alpha_2 < 1$ such that for any $\alpha > \alpha_2$ a run of SD on $\varphi' \circ \psi$ is whp $(\gamma n^\alpha, 2n^\alpha, d)$-safe.*

Proof: Let α_1 be obtained by application of Lemma 15 to φ'. Let $\alpha_2 = \max(\alpha_1, \eta)$. Then for $\alpha > \alpha_2$ whp run of SD on φ' is $(\gamma n^\alpha, n^\alpha, d)$-safe. Since for n large enough ψ contains less than n^α variables run of SD will be $(\gamma n^\alpha, 2n^\alpha, d)$-safe on $\varphi' \wedge \psi$. $\square$

Lemma 17 *Let $(D_0, \ldots, D_l)$ be an integer random process, $d > 0$, and let L, H be integer constants such that*

(a) $D_0 = 0, 0 < L < H$,

(b) $|D_{\tau+1} - D_\tau| = 1$,

(c) if $L \le D_\tau \le H$ the expectation of $D_{\tau+1}$ conditional to D_τ satisfies the inequality $\mathbf{E}\left(D_{\tau+1}|D_\tau\right) < D_\tau - d$ holds.

Then the probability that there is τ such that $D_\tau > H$ is less than $l \cdot e^{-d\frac{H-L}{2}}$.

Proof: We define a set of auxiliary processes D_τ^ξ:

$$D_\tau^\xi = \begin{cases} L, & \text{if } \tau < \xi, \\ D_\tau, & \text{if } (\tau \ge \xi),\ (D_\xi = L) \text{ and } (D_\zeta \ge L),\ \text{for all } \zeta \in \{\xi, \ldots, \tau\}), \\ D_\zeta - d(\tau - \zeta), & \text{if } \tau > \xi,\ D_\xi = L, \text{ and } \zeta \in \{\xi, \ldots, \tau\} \text{ is the least such that } D_\zeta < L, \\ L - d(\tau - \xi), & \text{otherwise, i.e., } D_\xi \ne L \text{ and } \tau \ge \xi. \end{cases}$$

The processes $D_\tau^0, \ldots, D_\tau^l$ are designed so that every D_τ^ξ for $\tau \ge \xi$ satisfies inequality $\mathbf{E}\left(D_{\tau+1}^\xi|D_\tau^\xi\right) \le D_\tau^\xi - d$. Indeed, suppose that $\tau \ge \xi$. If $D_\xi \ne L$ then

$$\mathbf{E}\left(D_{\tau+1}^\xi|D_\tau^\xi\right) = L - d(\tau + 1 - \xi) = (L - (\tau - \xi) - d = D_\tau^\xi - d.$$

Let $D_\xi = L$. If $D_\zeta \ge L$ for all $\zeta\{\xi, \ldots, \tau\}$ then $D_\tau^\xi = D_\tau$, $D_{\tau+1}^\xi = D_{\tau+1}$, and the result follows from the assumption $\mathbf{E}\left(D_{\tau+1}|D_\tau\right) < D_\tau - d$. If there is $\zeta \in \{\xi, \ldots, \tau\}$ with $D_\zeta < L$ then

$$\mathbf{E}\left(D_{\tau+1}^\xi|D_\tau^\xi\right) = \mathbf{E}\left(D_{\tau+1}^\xi|D_\zeta\right) = D_\zeta - d(\tau + 1 - \zeta) = (D_\zeta - d(\tau - \zeta)) - d = D_\tau^\xi - d.$$

By Azuma's inequality (1) for each ξ the probability of the event "there exists τ such that $D_\tau^\xi = H$" is less than $e^{-(H-L)d}$.

On the other hand let $D_\tau > L$ and ξ be equal to the number of the most recent step for which $D_\xi = L$. It is easy to see that $D_\tau = D_\tau^\xi$. Thus if at some step $D_\tau = H$ then there is $\xi < \tau$ such that $D_\tau^\xi = H$. Using the union bound we get the required inequality. □

Lemma 18 *Let $\varrho > \kappa' \cdot \ln n$ for some κ'. Let φ be a random 3-CNF sampled according to distribution $\Phi_\eta^{\mathtt{plant}}(n, \varrho n)$ such that run of SD on φ is whp $(\gamma_1 n^\alpha, \gamma_2 n^\alpha, 1)$-safe for some constants γ_1, γ_2 with $\gamma_1 > 3\gamma_2$. Let $\vec{u}_d(m), \vec{u}_l(m)$ denote the pair of assignments produced by the pair of processes (SD,LS) on step m. For any t, whp the Hamming distance between $\vec{u}_d(t)$ and $\vec{u}_l(t)$ does not exceed $\gamma_1 n^\alpha$.*

Proof: Let N_t be the set of tuples at Hamming distance at most $\gamma_1 n^\alpha$ from $\vec{u}_d(t)$, and $\mathcal{E}$ be event "$\vec{u}_l(t) \notin N_t$ for some t". LS starts with the same initial assignment as SD and we will show that it does not leave N_t.

At some steps the distance between $\vec{u}_d(t)$ and $\vec{u}_l(t)$ remains the same, and at some it changes. Let $\vec{u}_d, \vec{u}_l$ be the assignments produced by the algorithms after τ changes have taken place, and D_τ be the distance between them. If $2\gamma_2 n^\alpha < D_\tau < \gamma_1 n^\alpha$ we have $\mathbf{E}\left(D_{\tau+1}|D_\tau\right) < D_\tau - \frac{1}{3}$. Indeed, the number of variables voted to be zero does not exceed $\gamma_2 n^\alpha$ and is at least twice less than number of variables that differ in $\vec{u}_d(t)$ and $\vec{u}_l(t)$. Since any change in the distance between the assignments happens if and only if a variable voted to be 0 or a variable at which $\vec{u}_d(t)$ and $\vec{u}_l(t)$ are different, we have the required inequality. Now we can apply Lemma 17 for D setting $L = 2\gamma_2 n^\alpha, H = 3\gamma_2 n^\alpha, d = 1/3$ and get that probability of LS leaving N_t is less than $\varrho n e^{-n^\alpha/6}$. □

Corollary 19 *For $\varphi \in \Phi_\eta^{\mathtt{plant}}(n, \varrho n)$ there is a constant α_3 such that distance between $\vec{u}_d(t)$ and $\vec{u}_l(t)$ defined in Lemma 18 whp does not exceed n^{α_3}.*

We say that a variable *plays d-righteously in a run of LS* if every time it is considered for flipping it is d-righteous. Combining corollaries 16 and 19 we obtain the following

Lemma 20 *For any d there is $\alpha_4 < 1$ such that, for a run of LS on $\varphi \in \Phi_\eta^{\mathtt{plant}}(n, \varrho n)$ whp the number of variables that do not play d-righteously is bounded above by n^{α_4}.*

Proof: From Corollaries 16 and 19 it follows that whp at every step of LS the number of variables that are not d-righteous is less than $n^{\tilde{\alpha}}$, for some $\tilde{\alpha}$.

Therefore denoting the number of different assignments considered by LS by T (note that $T \le \varrho n$) and observing that at each step the probability to consider a variable voted to be 0 is $n^{\tilde{\alpha}-1}$ we obtain the following upper bound for the expectation of the number of non-d-righteous variables throughout the run:

$$Tn^{\tilde{\alpha}-1} \le \kappa' n(\ln n)n^{\tilde{\alpha}-1} = \kappa' n^{\tilde{\alpha}} \ln n \le n^{\tilde{\alpha}+\varepsilon}$$

for arbitrary ε with $\tilde{\alpha} + 2\varepsilon < 1$. We apply Markov inequality and obtain $\mathbf{P}\left(I > n^{\tilde{\alpha}+2\varepsilon}\right) \le n^{-\varepsilon}$, where I denotes the number of variables that do not play d-righteously. Now α_4 can be set to be $\tilde{\alpha} + 2\varepsilon$. □

A clause $(\overline{x}, \overline{y}, z)$ is called a *cap support* if there are w_1, w_2 such that (x, w_1, w_2, y, z) is a cap in φ. For a formula ψ we denote the set of variables that occur in it by $\mathrm{var}(\psi)$. For a set of clauses K we denote by $\bigwedge K$ a CNF formula constructed by conjunction of the clauses. For the sake of simplicity we will write $\mathrm{var}(K)$ instead of $\mathrm{var}\left(\bigwedge K\right)$. In what follows it will be convenient to view a CNF as a sequence of clauses. Note that representation of a CNF is quite natural when we sample a random CNF by generating random clauses. This way every clause occupies certain position in the formula. For a set of positions P we denote the formula obtained from φ by removing all clauses except for occupying positions P by $\varphi\downarrow_P$. The set of variables occurring in the clauses in positions in P will be denoted by $\mathrm{var}(P)$.

We denote by $\mathcal{C}$ the set of all possible clauses over n variables. Let us fix a real constant $\nu < 1$. We will need the following notation:

- let $[k]$ denote the set of the first k positions of clauses in φ, V be the set of all variables in φ;
- let $S^{\varphi,\nu}$ be the set of positions from $[n^\nu]$ occupied by clauses that are cap supports in φ, and $L^{\varphi,\nu}$ the set of variables that occur in clauses in positions $S^{\varphi,\nu}$;
- let $T^{\varphi,\nu}$ be set of positions of φ occupied by clauses containing a variable from $L^{\varphi,\nu}$;
- let $U^{\varphi,\nu}$ be the set of positions in φ occupied by clauses containing a variable from $\mathrm{var}\left(\varphi\downarrow_{[n^\nu]\setminus S^{\varphi,\nu}}\right)$;

- finally, let $R^{\varphi,\nu} = [\varrho n] \setminus (S^{\varphi,\nu} \cup U^{\varphi,\nu})$;
- let also $M^{\varphi,\nu} = \mathrm{var}(T^{\varphi,\nu})$ and $N^{\varphi,\nu} = \mathrm{var}(U^{\varphi,\nu})$.

Fig. 4 pictures the notation just introduced.

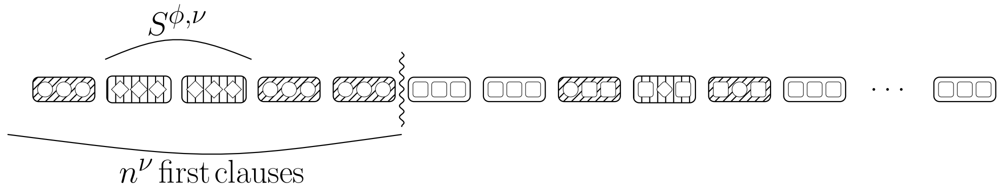

Figure 4: A scheme of a 3-CNF. Every clause is shown as a rectangle with its literals represented by squares inside the rectangle. Literals corresponding to variables from $L^{\phi\nu}$ and from $\mathrm{var}\left(\varphi\downarrow_{[n^{\nu}]\setminus S^{\varphi,\nu}}\right)$ are shown as diamonds and circles, respectively. Shaded rectangles with vertical and diagonal lines represent clauses from $T^{\phi\nu}$ and $U^{\phi\nu}$, respectively.

Lemma 21 *If $\rho \leq \kappa \ln n$ and $\kappa < \frac{7}{6}$ then there is μ_0 such that for any $\mu < \mu_0$ there is $\nu < 1$ such that whp:*

(1) $|S^{\varphi,\nu}| \sim n^{\mu}$*;*

(2) $M^{\varphi,\nu} \cap N^{\varphi,\nu} = \varnothing$*, that is variables from clauses from $U^{\varphi,\nu}$ do not appear in the same clauses with variables from $S^{\varphi,\nu}$;*

(3) $|M^{\varphi,\nu}| = 3|T^{\varphi,\nu}|$*, that is no variable occurs twice in the clauses from $T^{\varphi,\nu}$.*

Proof: It follows from Lemma 9 that for $\varrho \leq \kappa \ln n, \kappa < \frac{7}{6}$ there exists $\alpha, 0 < \alpha < 1$ such that the number of caps in the formula is $\sim n^{\alpha}$. We set

$$\mu_0 = \alpha/2, \qquad \nu = \mu + 1 - \alpha.$$

(1) For a subset R of all positions of clauses in ϕ let $\mathcal{C}_R$ denote event “R is exactly the set of positions occupied by cap supports”. Obviously for any sets $R_1, R_2, |R_1| = |R_2|$ we have $\mathbf{P}(\mathcal{C}_{R_1}) = \mathbf{P}(\mathcal{C}_{R_2})$. Thus positions of the cap supports are selected uniformly at random without repetition. By straightforward computation we have expectation of the number of cap supports among first n^{ν} clauses equal approximately $n^{\alpha} \cdot n^{\nu-1} = n^{\mu+1-\alpha-1+\alpha} = n^{\mu}$ and variance is bounded above by the expectation, so it follows from Chebyshev inequality that random variable “number of cap supports among first n^{ν} clauses” is whp $\sim n^{\mu}$.

(2) By Lemma 2(2) whp there is no variable that occurs in more than $\ln^2 n$ clauses. Therefore $|M^{\varphi,\nu}| = O(n^{\mu} \ln^2 n)$ and $|N^{\varphi,\nu}| = O(n^{\nu} \ln^2 n)$. These sets are randomly chosen from an n-element set, and therefore the probability they have a common element is at most $n^{\mu+\nu-1} \ln^4 n$. Due to definition of μ and ν we have $\mu + \nu - 1 < \alpha/2 + \alpha/2 + 1 - \alpha - 1 = 0$.

(3) Since whp $|T^{\varphi,\nu}| = O(n^{\mu} \ln^2 n)$, the probability that two clauses from this set share a variable is bounded above by $n^{2\mu-1} \ln^4 n$. We have $2\mu - 1 < \alpha - 1 < 0$ so this probability tends to 0. $\square$

Let us fix a formula φ selected accordingly $\Phi^{\mathsf{plant}}(n, \varrho n)$ and $\mu < \frac{1}{5}$, and let ν correspond to μ as in Lemma 21. Let T_0 and U_0 be subsets of $[\varrho n]$ such that $T_0 \cap U_0 = \varnothing$, $[n^{\nu}] \subseteq T_0 \cup U_0$ and let $S_0 = T_0 \cap [n^{\nu}]$. We denote by $H_{T_0 U_0}$ a hypothesis stating that φ is such that $S^{\varphi,\nu} = S_0$, $T^{\varphi,\nu} = T_0$, $U^{\varphi,\nu} = U_0$ and also $M^{\varphi,\nu} \cap N^{\varphi,\nu} = \varnothing$, $|M^{\varphi,\nu}| = 3\,|T^{\varphi,\nu}|$.

Lemma 22 *If for an event E there is a sequence $\delta(n) \underset{n\longrightarrow\infty}{\longrightarrow} 0$ such that for all pairs (T_0, U_0), $|T_0 \cup U_0| < n^{2\nu}$ we have $\mathbf{P}\left(E|H_{T_0U_0}\right) \leq \delta(n)$ then $\mathbf{P}\left(E\right) \underset{n\longrightarrow\infty}{\longrightarrow} 0$.*

Proof: We can bound probability of event E as

$$\begin{aligned} \mathbf{P}(E) &\leq \sum_{T_0,U_0:|T_0\cup U_0|<n^{2\nu}} \left(\mathbf{P}\left(E|H_{T_0U_0}\right)\mathbf{P}\left(H_{T_0U_0}\right) \right. \\ &\qquad \left. +\mathbf{P}\left(M^{\varphi,\nu}\cap N^{\varphi,\nu}\neq\varnothing \text{ or } |M^{\varphi,\nu}|<3\left|T^{\varphi,\nu}\right| \text{ or } |T_0\cup U_0|\geq n^{2\nu}\right)\right) \\ &\leq \delta(n)+\mathbf{P}\left(M^{\varphi,\nu}\cap N^{\varphi,\nu}\neq\varnothing\right)+\mathbf{P}\left(|M^{\varphi,\nu}|<3\left|T^{\varphi,\nu}\right|\right)+\mathbf{P}\left(|T_0\cup U_0|\geq n^{2\nu}\right). \end{aligned}$$

By Lemma 21 probabilities of events $M^{\varphi,\nu}\cap N^{\varphi,\nu}\neq\varnothing$ and $|M^{\varphi,\nu}|<3\left|T^{\varphi,\nu}\right|$ tend to 0 as n approaches infinity. By Lemma 2 (2) we have $|T_0\cup U_0|<n^{2\nu}$ whp. Thus we obtain the result. □

Observation 23 *If φ is selected according to $\Phi^{\mathsf{plant}}(n, \varrho n)$ conditioned to $H_{T_0U_0}$ then formula*

$$\varphi\downarrow_{[\varrho n]\setminus(T_0\cup U_0)}$$

has the same distribution as if it was generated by picking clauses from all clauses over variables $V\setminus\mathrm{var}([n^\nu])$ uniformly at random.

Proof: Let $\mathcal{C}'$ be the set of all clauses over variables in $V\setminus\mathrm{var}([n^\nu])$ and $R_0=[\varrho n]\setminus(T_0\cup U_0)$. Take a formula ψ such that positions from R_0 of this formula are occupied by clauses from $\mathcal{C}'$. It suffices to observe that the number of formulas ψ' such that $\psi'\downarrow_{R_0}=\psi\downarrow_{R_0}$, $S^{\psi',\nu}=S_0$, $T^{\psi',\nu}=T_0$, $U^{\psi',\nu}=U_0$ is the same for any ψ. So since all possible formulas over variables from some set are equiprobable a random formula is generated by random sampling of clauses. □

Proof: [of Proposition 12] We will bound probability of success of Local Search under a hypothesis of the form $H_{T_0U_0}$ and apply Lemma 22 to get the result. Let α_4 be the exponent corresponding to ϱ by Lemma 20, and choose μ and ν such that $\alpha_4+2\mu<1$.

Let $M=M^{\varphi,\nu}$ and $L=L^{\varphi,\nu}$. We split formula φ into $\varphi_1=\varphi\downarrow_{T_0}$ and $\varphi_2=\varphi\downarrow_{[\varrho n]\setminus T_0}$ and first consider a run of LS applied to φ_2 only. Formula φ_2 can in turn be considered as the conjunction of $\varphi_{21}=\varphi\downarrow_{U_0}$ and $\varphi_{22}=\varphi\downarrow_{[\varrho n]\setminus(T_0\cup U_0)}$. In Fig. 4 formula φ_1 consists of clauses shaded with vertical lines, formula φ_{21} of clauses shaded with diagonal lines and formula φ_{22} of clauses that are not shaded. By Observation 23 formula φ_{22} is sampled according to

$$\Phi^{\mathtt{plant}}(n-\delta_1(n), n\varrho-\delta_2(n))$$

modulo names of variables where $\delta_1(n)$ and $\delta_2(n)$ are $o(n)$. So formula φ_2 is sampled according to

$$\Phi^{\mathtt{plant}}_{2\mu}(n-\delta_1(n), n\varrho-\delta_2(n)).$$

By Lemma 20 the number of variables that do not play 2-righteously during run of LS on φ_2 is bounded from above by n^{α_4} for a certain $\alpha_3<1$.

We consider coupling $(LS_\varphi, LS_{\varphi_2})$ of runs of LS on φ and φ_2, denoting assignments obtained by the runs of the algorithm at step t by $\vec{u}_\varphi(t)$ and $\vec{u}_{\varphi_2}(t)$ respectively. Let K be the set of those variables which do not belong to L (squares and circles in Fig. 4). Formula φ_2 is a 3-CNF containing only variables from

K. For an assignment of values of all variables $\vec{u}$ we will denote by $\vec{u}|_K$ its restriction onto variables from K. We make process LS_φ start with a random assignment $\vec{u}_\varphi(0) = \vec{u}^0_\varphi$ to all variables, and LS_{φ_2} with a random assignment $\vec{u}_{\varphi_2}(0) = \vec{u}^0_{\varphi_2}$ to variables in K, such that $\vec{u}^0_\varphi|_K = \vec{u}^0_{\varphi_2}$. Now the algorithms work as follows. At every step a random variable x_i is chosen. Process LS_φ makes its step, and process LS_{φ_2} makes its step if $x_i \in K$.

Whp LS_{φ_2} will run with at most n^{α_4} variables that do not play 2-righteously. Let W denote the set of such variables. Variables in formula φ_1 are selected uniformly at random so if $\alpha_4 + 2\mu < 1$ then whp set M does not intersect with W. Hence, every time LS_φ considers some variable from M it is 2-righteous in φ_2 and belongs to at most one clause of φ_1. Therefore such a variable is at least 1-righteous φ and is flipped to 1, or stays 1, whichever is to happen for LS_{φ_2}. Thus whp at every step of $(LS_\varphi, LS_{\varphi_2})$ we have $\vec{u}_\varphi(t)|_K = \vec{u}_{\varphi_2}(t)$. In the rest of the proof we consider only this highly probable case.

Consider some cap support $c_i = (\overline{x}_1, \overline{x}_4, x_5)$ occupying a position $i \in [n^\nu]$ and such that $x_1 = 0, x_4 = 1, x_5 = 0$ at time 0, and a set P_{c_i} of variables occurring in clauses that contain variables $\mathrm{var}(c_i)$ (obviously $\mathrm{var}(c_i) \subseteq P_{c_i}$). Let c_j be the clause that forms a cap with c_i. We say that a variable is *discovered* at step t if it is considered for the first time at step t. Let $p_1, \dots, p_k$ be an ordering of elements of P_{c_i} according to the step of their discovery. In other words if variable p_1 is the first variable from P_{c_i} that is discovered, p_k was the last. In the case some variables are not considered at all, we place them in the end of the list in a random order. Observe that all variables that play at least 1-righteously are discovered at some step. All orderings of variables are equiprobable, hence, the probability of variables $\mathrm{var}(c_i)$ to occupy places p_{k-2}, p_{k-1} and p_k equals $3!/k(k-1)(k-2)$. We will call this ordering *unlucky*.

Let us consider what happens if the order of discovery of P_{c_i} is unlucky. All variables in $P_{c_i} \setminus \mathrm{var}(c_i)$ play 1-righteously, therefore once they are discovered by LS_φ they equal to 1. Thus when x_1, x_4, x_5 are finally considered all clauses they occur in are satisfied, except for c_j. So variables x_1, x_4, x_5 do not change their values and the clause c_j remains unsatisfied by the end of the work of LS_φ.

By Lemma 2(2) whp no vertex has degree greater than $\ln^2 n$, so the size of the set P_{c_i} is bounded above by $3\ln^2 n$. Thus the probability of event $Unluck(i)$ =“order of discovery of $\mathrm{var}(c_i)$ is unlucky” is greater than $\frac{1}{\ln^6 n}$. Thus, the expectation of $|\{i|Unluck(i)\}|$ equals

$$\frac{|S_0|}{\ln^6 n} = \frac{n^\mu}{\ln^6 n}.$$

Any variable whp occurs in clauses from $T^{\varphi,\nu}$ at most once, hence there is no variable that occurs in the same clause with a variable from c_{i_1} and a variable from c_{i_2} for $i_1, i_2 \in S_0$, $i_1 \neq i_2$. This implies that events of the form $Unluck(i)$ are independent. Therefore random variable $|\{i|Unluck(i)\}|$ is Bernoulli and, as its expectation tends to infinity, the probability that it equals to 0 goes to 0. Since unlucky ordering of at least one cap support leads to failure of the LS this proves the result. $\square$

References

[1] D. Achlioptas. Lower bounds for random 3-SAT via differential equations. *Theor. Comput. Sci.*, 265(1-2):159–185, 2001.

[2] D. Achlioptas and E. Friedgut. A sharp threshold for k-colorability. *Random Struct. Algorithms*, 14(1):63–70, 1999.

[3] M. Alekhnovich and E. Ben-Sasson. Linear upper bounds for random walk on small density random 3-cnfs. *SIAM J. Comput.*, 36(5):1248–1263, 2007.

[4] N. Alon and J. Spencer. *The Probabilistic Method*. John Wiley, 2000.

[5] E. Amiri and E. Skvortsov. Pushing random walk beyond golden ratio. In *CSR*, pages 44–55, 2007.

[6] E. Ben-Sasson, Y. Bilu, and D. Gutfreund. Finding a randomly planted assignment in a random 3-cnf. Manuscript, 2002.

[7] A. Braunstein, M. Mézard, and R. Zecchina. Survey propagation: An algorithm for satisfiability. *Random Struct. Algorithms*, 27(2):201–226, 2005.

[8] A. Bulatov and E. Skvortsov. Efficiency of local search. In *SAT*, pages 297–310, 2006.

[9] H. Chen. An algorithm for sat above the threshold. In *SAT*, pages 14–24, 2003.

[10] J. M. Crawford and L. D. Auton. Experimental results on the crossover point in random 3-sat. *Artif. Intell.*, 81(1-2):31–57, 1996.

[11] O. Dubois, Y. Boufkhad, and J. Mandler. Typical random 3-sat formulae and the satisfiability threshold. *CoRR*, cs.DM/0211036, 2002.

[12] U. Feige, E. Mossel, and D. Vilenchik. Complete convergence of message passing algorithms for some satisfiability problems. In *APPROX-RANDOM*, pages 339–350, 2006.

[13] A. Flaxman. A spectral technique for random satisfiable 3cnf formulas. In *SODA*, pages 357–363, 2003.

[14] J. Franco and Paull M. Probabilistic analysis of the davis-putnam procedure for solving satisfiability. *Discr. Appl. Math.*, 5:77–87, 1983.

[15] J. Gu. Efficient local search for very large-scale satisfiability problem. *ACM SIGART Bulletin*, 3(1):8–12, 1992.

[16] P. Hansen and B. Jaumard. Algorithms for the maximum satisfiability problem. *Computing*, 44:279–303, 1990.

[17] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[18] A. Kaporis, L. Kirousis, and E. Lalas. The probabilistic analysis of a greedy satisfiability algorithm. In *ESA*, pages 574–585, 2002.

[19] E. Koutsoupias and C. Papadimitriou. On the greedy algorithm for satisfiability. *Inf. Process. Lett.*, 43(1):53–55, 1992.

[20] M. Krivelevich, B. Sudakov, and D. Vilenchik. On the random satisfiable 3cnf process. submitted, 2008.

[21] M. Krivelevich and D. Vilenchik. Solving random satisfiable 3cnf formulas in expected polynomial time. In *SODA*, pages 454–463, 2006.

[22] M. Mézard, T. Mora, and R. Zecchina. Clustering of solutions in the random satisfiability problem. *CoRR*, abs/cond-mat/0504070, 2005.

[23] D. Mitchell. A sat solver primer. *Bulletin of the EATCS*, 85:112–132, 2005.

[24] Christos H. Papadimitriou. On selecting a satisfying truth assignment (extended abstract). In *FOCS*, pages 163–169, 1991.

[25] B. Selman, H. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. In *AAAI*, pages 440–446, 1992.

[26] N. Wormald. Differential equations for random processes and random graphs. *The Annals of Applied Probability*, 5(4):1217–1235, 1995.