Department of Computer Science and Applied Mathematics
Weizmann Institute of Science

Planted Random 3SAT with a Small Fraction of 1-Clauses

Submitted for the degree of Master of Science
to the Scientific Council of the Weizmann Institute of Science

Under the supervision of Professor Uriel Feige

Alina Arbitman

February 2012

Abstract

A planted random 3SAT instance is formed by selecting a truth assignment and including each clause consistent with it with a certain probability. When the clause to variable ratio is $\Omega(\log n)$ it is well known that the assignment can be reconstructed by the Majority Vote heuristic, hence the more interesting case is when the clause to variable ratio is constant. In a paper from 2003, Flaxman presented a modified version of the planted model where clauses satisfied by different number of literals are included with different probabilities. We focus on the case where the number of clauses satisfied by exactly one literal is small, both in absolute and in relative terms. We present polynomial time algorithms for the two distribution families, where for the first one we are also able to handle a semi random model for choosing the polarities of the formula.

Acknowledgments

First and foremost, I would like to express my deep gratitude to my advisor, Uriel Feige, for outstanding guidance and encouragement, for introducing me with his lucid way of thinking and for a great sense of humour.

Many thanks to the Mathematics and Computer Science faculty, and in particular to Oded Goldreich for enriching courses and reading material and Itai Benjamini for inspiring courses and conversations.

Thanks to my fellow students at the faculty, for making each day at Weizmann a true enjoyment.

Last but not least, I would like to thank my family and my dear Maor, for their endless love and support.

1 Introduction

The classic problem of $3SAT$ is concerned with finding a satisfying assignment to an input 3CNF formula in polynomial time. A 3CNF formula over the Boolean variables $x_1, ..., x_n$ is the conjunction of m clauses $c_1, ..., c_m$, where each clause is the disjunction of 3 literals, $c_i = \ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}$, and each literal $\ell_{i,j}$ is either a variable or its negation (we denote the negation of x by $\sim x$). A $3CNF$ formula is satisfiable if there is an assignment of variables to $\{True, False\}$ so that every clause contains at least one literal assigned $True$. The 3SAT problem is well known to be $\mathcal{NP}$-Complete and no algorithm can succeed on all 3SAT instances in polynomial time, unless $\mathcal{P} = \mathcal{NP}$ ([11],[22]). The intractability of 3SAT in the worst case has lead to an extensive average case research. We concentrate on a probabilistic model for generating random 3CNF instances named 'planted random 3SAT', and present algorithms for a previously suggested variant of that model.

A planted random instance of 3SAT is formed by selecting a truth assignment ϕ on n variables uniformly at random, and then including each clause satisfied by the planted assignment ϕ with probability p. In a paper from 2003 Flaxman extended this model, suggesting to assign different probabilities according to the number of literals in the clause that are satisfied by ϕ ([16]). Let p_i be the probability to include a clause with exactly i literals satisfied by ϕ. Flaxman showed that for any constants $\eta_2, \eta_3 \in [0, 1]$ there is a constant d_{min} such that for all $d > d_{min}$ his spectral algorithm finds a satisfying assignment whp over instances with $p_1 = \frac{d}{n^2}$, $p_2 = \frac{\eta_2 d}{n^2}$ and $p_3 = \frac{\eta_3 d}{n^2}$. The algorithm relies on there being some positive fraction of clauses with exactly one satisfied literal and does not handle the case where $p_1 < \min\{p_2, p_3\}$. We extend Flaxman's result by addressing two separate ranges in which $p_1 < \min\{p_2, p_2\}$:

1. $p_2 + p_3 = \frac{d}{n^2}$ and $p_1 \leq \frac{c}{(d \log d)n^2}$ for a universal constant c and sufficiently large d.

   For somewhat smaller values of p_1 we are also able to deal with a semi-random version of the model.

2. $p_1 = \frac{d_1}{n^2}$ and $p_2 + p_3 = \frac{d_2}{n^2}$ with $2d_1 \leq d_2 \leq 2^{cd_1}$ for a universal constant c and sufficiently large d_1.

   The leading constant in $2d_1 \leq d_2$ is to some extant arbitrary[1].

Next we define the exact models in use and state the main result.

1.1 The Model

Definition 1 (i-clause)**.** *For $i \in \{0, 1, 2, 3\}$ we say a clause c is an **i-clause** w.r.t assignment ψ (to the variables that appear in c) if the number of literals in c satisfied by ψ is i. For $i \in \{1, 2, 3\}$ we may also say that ψ **satisfies** c as an i-clause.*

We consider two models for generating $3SAT$ instances:

Definition 2 (**Planted Random 3SAT Model).** *In this model an instance of 3SAT is formed by first choosing a truth assignment on n variables uniformly at random, and then selecting each i-clause for $i \in \{1, 2, 3\}$ independently with probability p_i, for some $p_i \in [0, 1]$.*

Definition 3 (**Planted Semi-Random 3SAT Model).** *In this model an instance of 3SAT is formed by first choosing a truth assignment on n variables uniformly at random and then selecting m clauses as follows: for each clause its variables are chosen independently at random; then their polarities may be chosen adversarially, as long as all clauses are satisfied and the number of 1-clauses does not exceed ϵm.*

Note: There are two selection processes underlying the models described above: in one, every i-clause is selected independently with an appropriate probability; in the other, m clauses are selected independently from all legal triples of literals (with an appropriate fraction of i-clauses for every i). We shall use the two

[1]For most values of p_1, p_2 and p_3 for which $d_2 \leq 2d_1$ the analysis of Flaxman's algorithm applies, and hence our work does not address this range of parameters. There is a range of parameters with $d_2 \leq 2d_1$ where neither Flaxman's algorithm nor ours work, namely, when $p_1 = p_3$ and p_2 is small. For such values the bottleneck is in finding an approximate assignment (the first stage of the algorithm).

processes interchangeably throughout the work, for the sake of simplicity of the presentation in question.

1.2 Our Result

We shall later introduce three algorithms: $Alg1$, $Alg2$ and $Alg3$; regarding these algorithms we prove the following results.

Theorem 1. *There exists a constant $c < 1$ such that for every sufficiently large d the following holds: let F be a planted random 3SAT formula generated according to the random model with $p_1 \leq \frac{c}{(d \log d)n^2}$ and $p_2 + p_3 = \frac{d}{n^2}$, then the $Alg1$ algorithm finds a satisfying assignment to F in polynomial time whp over the choice of F.*

Theorem 2. *There exists a constant $c < 1$ such that for every sufficiently large d the following holds: let F be a planted random 3SAT formula generated according to the semi-random model with $m = dn$ and $\epsilon \leq \frac{c}{d^4 \log^2 d}$, then the $Alg2$ algorithm finds a satisfying assignment to F in polynomial time whp over the choice of F.*

Remark: Theorem 2 still holds even if F contains also 0-clauses ($w.r.t$ the planted assignment), as long as the number of 0- and 1- clauses together does not exceed ϵm.

Theorem 3. *There exists a constant $c < 1$ such that for every sufficiently large d_1 and for d_2 with $2d_1 \leq d_2 \leq 2^{cd_1}$ the following holds: let F be a planted random 3SAT formula generated according to the random model with $p_1 = \frac{d_1}{n^2}$ and $p_2 + p_3 = \frac{d_2}{n^2}$, then the $Alg3$ algorithm finds a satisfying assignment to F in polynomial time whp over the choice of F.*

1.3 Related Work and Motivation

In the Random $3SAT$ problem a 3CNF formula on n Boolean variables is generated by selecting m random clauses independently and uniformly from all triples of literals. The goal is to find an assignment of variables to truth values so that the entire formula is satisfied, or prove that no such assignment exists. The probability

of an instance drawn from this distribution to be satisfiable has an interesting connection to m/n, which is the clause to variable ratio (often also referred as 'the clause density'). When the ratio is low, the instances are likely to be satisfiable, whereas when the ratio is high, the instances are unsatisfiable with high probability. This satisfiability threshold is known to lie between 3.42 [19] and 4.5 [20] and experiments suggest that it is approximately 4.2 [12]. This is not yet proven, but Friedgut [18] has shown that there exists a sequence γ_n such that if $m/n \leq \gamma_n - \epsilon$ the probability of an instance (with such density) to be satisfiable tends to 1 and if $m/n \geq \gamma_n + \epsilon$ this probability tends to 0. It is still not known, however, if the sequence γ_n converges (or, equivalently, can be taken as a constant threshold). Several algorithms are known to perform well for instances drawn from the random $3SAT$ distribution with low clause density. One such algorithm is the *Pure Literals Heuristic*, for which a density as low as $d \approx 1.63$ suffices ([23]). Another greedy algorithm succeeds with asymptotically positive probability for $d \approx 3.42$ ([19]). Furthermore, experimental results suggest that algorithms from the *Survey Propagation* family succeed for density very close to the conjectured satisfiability threshold ([8]).

When evaluating algorithms, we generally require they perform well on most instances drawn from a certain distribution. For instances with density above the threshold, this means we expect them to find satisfying assignments for a small fraction of the probability space, as only such fraction of instances is satisfiable. It is thus natural to consider the conditional distribution, where a 3SAT instance is formed by first selecting a formula at random and then keeping it only if it is satisfiable. But unfortunately this distribution is both difficult to sample from and to analyse. This has lead to research on 'planted' random 3SAT. In the planted model first a truth assignment on n variables is selected uniformly at random (this assignment is referred to as 'the planted assignment') and then the formula is selected at random from all formulas which are satisfied by this assignment. One selection process which guarantees the resulting formula is indeed satisfied by the planted assignment is picking m clauses independently and uniformly at random only from all triples of literals which are satisfied by the planted assignment. In general, for a given triple of variables, there are $2^3 = 8$ ways of choosing the variables' polarities (for every variable x we take either x or its negation $\sim x$). If

we are constrained by an assignment chosen beforehand, then this number reduces to 7, since there is exactly one possible choice of polarities for which all literals are not satisfied by the assignment (resulting in the entire clause being unsatisfied), and that choice should not be included in the instance. Another natural selection process is choosing every clause satisfied by the assignment independently with a certain probability p (sometimes it may be more convenient to think of this process as proceeding in two stages: on the first stage we pick three variables at random and then select their polarities out of the 7 legal possibilities).

When comparing the planted distribution to the uniform distribution on satisfiable instances, we notice that the number of assignments an instance has increases its probability to be selected in the planted model, whereas in the uniform distribution clearly all satisfiable instances are selected with the same probability. For a clause density as high as $\Omega(\log n)$ a random satisfiable instance is likely to have only one satisfying assignment, which may explain why in that range the two distributions are statistically close ([5]). A justification to investigate the planted distribution for a constant density may be found in a recent work, showing that for such density (with sufficiently large constant) the two distributions possess a similar structure of solution space, implying that in many cases algorithms for the planted distribution may be applied for the uniform one as well ([10]).

Planted distributions have been the focus of research in several different contexts: planted graph coloring ([6],[3]), planted bisection ([7]), planted clique ([14], [4]) and planted 3SAT ([21],[16],[15]). Some of the works inspired ours, and especially those of Flaxman ([16]) and Alon and Kahale ([3]) whose techniques are applied here.

In the context of 3SAT, for instances drawn from the planted distribution with clause density as high as $\Omega(\log n)$, the Majority Vote reconstructs the planted assignment *whp* ([21]). This heuristic assigns to each variable the truth value that satisfies the majority of the clauses in which it appears. The motivation behind it the following: if every clause consistent with the planted assignment is included with the same probability, then there is a bias towards including the literal satisfied by the planted assignment more frequently than its negation. For lower densities (i.e., constant ones) the Majority Vote would not satisfy the entire formula but may serve as a good starting point, and then the k-opt heuristic can complete

the assignment, as demonstrated in [15]. When the probabilities to include each i-clause in the instance are set appropriately, the Majority Vote fails, but in such case spectral steps apply ([16]).

As mentioned, Flaxman suggested a generalized version of the planted model where clauses with different numbers of satisfied literals are included with different probabilities. A further appealing generalization considers semi-random models, wherein the underlying principle is in there being a mixture of random and adversarial elements, with a varying proportion of the two. In general, the larger the portion of the adversarial elements our algorithms can withstand, the larger the probability space treated by them is, and hence the more robust they are.

Semi-random models have been suggested and studied for several problems involving planted distributions, and among others, independent sets and graph bisections ([13]) and graph coloring ([6]). In the context of 3SAT, Vilenchik and Feige [15] considered an adversary who is allowed to add arbitrary 3-clauses to a previously generated planted random 3SAT instance with a constant clause density.

In our work one of the models addressed is a semi random one, in which first the variables of each clause are chosen randomly and then an adversary is allowed to choose their polarities, as long as the fraction of 1-clauses is small.

2 Definitions

Definition 4 (Support)**.** *We say a variable x* ***supports*** *a clause c w.r.t assignment ψ if c is a 1-clause w.r.t ψ and x (or $\sim x$) is the satisfying literal of c w.r.t ψ.*

Definition 5 (Partial assignment)**.** *Let $X = \{x_1, ..., x_n\}$ be a set of Boolean variables. A* ***partial assignment*** *to X is an element of $\{True, False, *\}^n$, where we regard a variable assigned $*$ as unassigned. We say a clause is satisfied by a partial assignment ψ (to the variables appearing in it) if it is satisfied by any of its literals assigned in ψ.*

Definition 6 (c-expansion)**.** *We say a set of k clauses has* ***c-expansion*** *if the number of distinct variables in the set is at least ck. In such case we may also say that the set is* ***c-expanding****.*

Definition 7 (Formula graph)**.** *We associate with a formula F the following natural graph: the vertices represent the variables and two vertices share an edge if there is a clause in F which contains their variables. We call this graph the* ***formula graph*** *of F (in fact it is a multi-graph).*

3 The Algorithms

Similar to the approaches in [3], [16], our algorithms proceed in three main steps: finding an approximate assignment, an 'Unassignment' phase and completing the partial assignment to one that satisfies all clauses. For each step we have several different procedures possible.

3.1 Approximation

Majority Vote Given a $3CNF$ formula F, for every variable x compute the Majority Vote for it, that is, count both the number of positive and negative occurrences of x in the formula. If the first counter is larger than the second one assign $True$ to x. Otherwise assign $False$.

Reduction to MAX2SAT Given a $3CNF$ formula F, reduce it to $MAX2SAT$ as follows:

1. Convert F to a $2CNF$ formula by transforming each clause, say, $(x \vee y \vee \sim z)$, to $(x \vee y) \wedge (x \vee \sim z) \wedge (y \vee \sim z)$;

2. Apply on the $2CNF$ formula an approximation algorithm for $MAX2SAT$ known to have the following guarantee: if the input formula is $(1-\epsilon)$-satisfiable then the assignment output by the algorithm will satisfy at least a $1 - O(\sqrt{\epsilon})$ fraction of all clauses. Then take the assignment returned by the algorithm applied. For example, the algorithm that was suggested by Charikar, Makarychev and Makarychev in 2009 has such guarantee ([9]).

3.2 Unassignment

0-Clause Unassignment Let ψ be the approximate assignment from the previous step.

While there are clauses unsatisfied by ψ,

1. Form a partial assignment ψ' from ψ by unassigning all variables that appear in such clauses.
2. $\psi \leftarrow \psi'$.

Consider the final partial assignment (after the last iteration), denote it by σ. Then we notice there are only two types of clauses $w.r.t$ σ: 'satisfied' clauses (in which at least one of the variables is assigned) and 'unassigned' clauses - clauses whose variables are all unassigned in σ.

Small-Support Unassignment Let again ψ be the approximate assignment from the previous step.

1. Form a partial assignment ψ' from ψ by unassigning all variables which support less than $\frac{d_1}{2}$ clauses $w.r.t$ ψ.
2. While there are variables which support less than $\frac{d_1}{3}$ fully assigned clauses $w.r.t$ ψ',
   (a) Form a partial assignment ψ'' from ψ' by unassigning all such variables.
   (b) $\psi' \leftarrow \psi''$.

3.3 Completing the partial assignment

Matching Let σ be the final partial assignment of the '0-Clause Unassignment' phase and consider the set of 'unassigned' clauses $w.r.t$ σ. Construct a bipartite graph for this set of clauses, as follows: on the left hand side we will have one vertex for each variable and on the right hand side one vertex for each clause. A left hand side vertex and a right hand side vertex share an edge only if the corresponding

variable appears in the corresponding clause. Find a maximum matching in the graph. If every right hand side vertex is matched in this matching, assign each variable according to the demand of the clause matched to it. Otherwise, fail. Note that a maximum matching in a bipartite graph may be found efficiently.

Exhaustive Search Let σ be the final partial assignment of the 'Small-Support Unassignment' phase. First simplify F as follows: set all variables assigned in σ according to their assignment, remove all satisfied clauses and remove the assigned variables from the remaining clauses. If this results in an empty clause, fail. Otherwise, consider the simplified formula yet to be handled. Notice it has three types of clauses: clauses with 1, 2 or 3 literals. Next perform a *Unit Propagation*, that is, apply iteratively the following: 1. If there is a clause with a single literal, set its variable as required by the polarity. 2. Simplify the formula as previously. 3. If this results in an empty clause, fail. At the end of this procedure we are left only with clauses of length 2 or 3. Consider the formula graph induced by the formula at hand. If it contains a connected component of size larger than $\log n$, fail. Otherwise look for a satisfying assignment by performing an exhaustive search over the variables in each connected component of this graph separately. If no assignment satisfies the formula, fail. Otherwise return the satisfying assignment.

3.4 The algorithms

The three algorithms proceed as follows:

Alg1

1. **Majority Vote**
2. **0-Clause Unassignment**
3. **Matching**

Alg2

1. **Reduction to MAX2SAT**

2. **0-Clause Unassignment**

3. **Matching**

Alg3

1. **Majority Vote**

2. **Small-Support Unassignment**

3. **Exhaustive Search**

3.5 Algorithms Overview

Our work addresses two different 3SAT distribution families: in one, the absolute number of 1-clauses is small ($p_1 \leq O(d \log d)^{-1} n^{-2}$), while in the other this number is small in relative terms ($p_1 = d_1 n^{-2}$ for $d_1 \leq d_2/2$). For the first distribution, we consider both a random and a semi-random model versions.

For the random model (in both distribution families), our algorithms begin by applying the democratic procedure which counts for each variable both the number of clauses in which it appears as a positive literal and those in which it appears negatively (it treats each such occurrence as a 'vote') and makes its decision based on the majority preference. A delicate choice of the probabilities to include each i-clause in the formula might fool the Majority Vote. That is the reason why Flaxman's algorithm begins with spectral steps instead. In our case, however, the fraction of 1-clauses is small enough for this heuristic to be applicable.

In the semi-random model an adversary chooses the polarities and so she can tilt the statistics of a large linear set of variables, causing their Majority Vote to be uninformative. We overcome this obstacle by exploiting the small number of 1-clauses in a different manner; we observe that the formula reduced to its 2- and 3-clauses is in fact a satisfiable 2SAT instance, and hence a MAX2SAT approximation algorithm may be applied on the entire formula. In such case the initial number of 1-clauses should be somewhat smaller than in the random model.

An interesting feature of the model version in which the fraction of 1-clauses is small in absolute terms is that all 1-clauses may be replaced by 0-clauses. In

such case the planted assignment itself does not satisfy the formula, but our proofs show that a satisfying assignment exists and moreover can be found in polynomial time.

For a clause density which does not depend on n (as we consider), the Majority Vote is not likely to satisfy the formula and therefore a correction is required; indeed in our algorithms it is followed by an unassignment procedure. For the range in which the number of 1-clauses is small in absolute terms ($p_1 \leq O(d \log d)^{-1} n^{-2}$), we consider a very natural iterative process which we call '0-Clause Unassignment': it begins by unassigning all variables that appear in clauses unsatisfied by the Majority Vote and does it iteratively until all clauses are satisfied by the obtained partial assignment. For a sufficiently small number of initial 1-clauses, the number of clauses whose variables would be unassigned during such process is small. This ensures, for an instance which was initially generated in a random manner, that the residual formula possesses a particular enpension property, which guarantees the existence of a clause to variable matching (by Hall's theorem). In such case we are able to complete the assignment by assigning each variable according to the preference of the clause matched to it.

When the number of 1-clauses is small only in relative terms ($d_1 \leq d_2/2$), however, the unassignment procedure described above does not have to result in a small number of unassigned clauses and therefore a different kind of unassignment is required. For such case we consider another natural procedure, inspired by [16], which we call 'Small-Support Unassignment': at every iteration all variables that do not support enough fully assigned clauses are unassigned. This procedure is based on 1-clauses and as such it has the following useful property: every variable which is assigned by the Majority Vote differently than by the planted assignment and has also 'survived' the unassignment must support many clauses, each of which contains another variable on which the two assignments disagree. Then the subgraph induced by such variables would have more edges than expected from a similar-sized subgraph in a random formula. Indeed it is this property that guarantees no variables on which the Majority Vote disagrees with the planted assignment 'survive' the unassignment phase.

To complete the assignment we consider the residual formula graph and perform a very simple procedure: exhaustive search over all connected components of this

graph. When the number of unassigned variables at the end of the previous step is small (as we are able to show), this graph does not contain any connected component larger than $\log n$ *whp*. Since the unassignment procedure is based on 1-clauses, whose number depends on d_1, whereas the connected components are induced by 2- and 3-clauses as well (whose number depends on d_2), here we need an additional assumption restricting the number of 2- and 3-clauses as a function of the number of 1-clauses: $d_2 \leq 2^{O(d_1)}$.

It is not clear to us how this assumption can be removed. Increasing the number of 2- and 3-clauses will improve the Majority Vote further, but might also result in larger than $\log n$-sized components in the residual formula graph. In such case a different procedure than an exhaustive search may be needed for completing the assignment.

4 Correctness

4.1 Approximation

Lemma 4.1. *Let F be a planted random 3SAT formula generated according to the random model with $p_1 = \frac{d_1}{n^2}$, $p_2 + p_3 = \frac{d_2}{n^2}$, where the actual parameters are as in Theorem 1 or 3. Then whp over the choice of F the Majority Vote disagrees with the planted assignment on at most $2^{-\Omega(d_2)}n$ variables.*

Proof. Take any variable x. First we show that the Majority Vote *maj* and the planted assignment ϕ disagree on its assignment with probability $2^{-\Omega(d_2)}$. Assume $\phi(x) = True$ and fix two more variables y and z. We are interested in the number of clauses consisting of the three variables x, y and z, where x appears as a positive literal (x) and in those where it appears negatively ($\sim x$). In total we have four such clauses with positive occurrences of x: one 1-clause, two 2-clauses and one 3-clause, and three clauses with negative occurrences: two 1-clauses and one 2-clause. Let p_x be a random variable counting the actual number of clauses in F where x appears positively and similarly n_x for negative occurrences. Then $E[p_x] = (p_1 + 2p_2 + p_3)n^2$ and $E[n_x] = (2p_1 + p_2)n^2$. The Majority Vote disagrees with ϕ on the assignment of x when $p_x \leq n_x$. Since $E[p_x] - E[n_x] = (p_2 + p_3 - p_1)n^2 = d_2 - d_1 \geq d_2/2$, where the last inequality is due to the assumed parameters in Theorem

1 or 3, and both p_x and n_x are binomial random variables, we have $Pr[p_x \leq (E[p_x] + E[n_x])/2] \leq 2^{-\Omega(d_2)}$ and similarly $Pr[n_x \geq (E[p_x] + E[n_x])/2] \leq 2^{-\Omega(d_2)}$. We conclude that $Pr[p_x \leq n_x] \leq 2^{-\Omega(d_2)}$. Next from linearity of expectation, the expected number of variables on which maj and ϕ disagree is $2^{-\Omega(d_2)}n$ and by Markov inequality this happens with probability $1 - 2^{-\Omega(d_2)}$.

A stronger concentration result (with probability that depends only on n) may be obtained by looking at the process of clauses selection as a martingale with bounded difference and applying Azuma's inequality; consider the process in which m clauses are selected independently at random. Let M_i denote the number of variables on which maj and ϕ disagree up to the selection of the i-th clause, for $i \in [1, m]$ (then M_m is the total number of variables on which maj and ϕ disagree). Since each new selected clause can effect the Majority Vote of at most 3 variables, it holds that $|M_i - M_{i+1}| \leq 3$ for all i. In such case Azuma's inequality guarantees that $Pr[|M_m - E[M_m]| \geq t] \leq 2exp\left(-\frac{t^2}{18m}\right)$. Plugging in $t = E[M_m]$ we obtain that with probability $1 - e^{-\frac{\Omega(n)}{2^{\Omega(d_2)}}}$, M_m is indeed $2^{-\Omega(d_2)}n$ as expected. ■

Lemma 4.2. *Let F be a planted random 3SAT instance as in Theorem 1. Then whp over the choice of F the number 0- and 1-clauses w.r.t the Majority Vote is at most $\frac{n}{O(d \log d)}$ (and the rest are 2- or 3-clauses).*

Proof. In the following analysis we consider the model in which $m = dn$ clauses are picked uniformly at random.

The 1-clauses of F ($w.r.t$ the planted assignment ϕ) may serve as 0- or 1-clauses $w.r.t$ the Majority Vote maj as well, and their number is $\frac{n}{O(d \log d)}$. Apart from these clauses, any 2- or 3-clause $w.r.t$ ϕ that maj and ϕ disagree on one or more of its variables might also contribute to the count of 1- and 0-clauses $w.r.t$ maj. Hence we are interested in bounding the number of such clauses. Look at all variables upon which maj and ϕ disagree. We would condition on the event there are $2^{-\alpha d}n$ such variables for some constant α, as guaranteed by Lemma 4.1. Let X be a fixed set of $2^{-\alpha d}n$ variables. The average degree of X may be expressed by $\frac{1}{|X|}\sum_{x \in X, c \in F} \mathbf{1}_{\{x \in c\}}$ (where the summation is over all clauses of F and variables of X and $\mathbf{1}_{\{x \in c\}}$ represents the indicator variable of the event 'x appears in the clause

c').

$$E\left[\frac{1}{|X|}\sum_{x\in X, c\in F}\mathbf{1}_{\{x\in c\}}\right] = \frac{1}{|X|}\sum_{x\in X, c\in F} Pr[x\in c] = \frac{1}{2^{-\alpha d}n}\cdot 2^{-\alpha d}n\cdot dn\cdot\frac{3}{n} = 3d.$$

Since the selection process of the clauses is independent, by Chernoff bound we know that for $\delta > \alpha/2$ with probability $1-2^{-\Omega(\alpha d 2^{-\alpha d})n}$ the average degree of X is at most $(1+\delta)3d$. Taking the union bound over all sets of size $2^{-\alpha d}n$, we obtain the probability that any such set has an average degree larger than $(1+\delta)3d$ is upper bounded by the following expression:

$$\binom{n}{2^{-\alpha d}n}2^{-\Omega(\alpha d 2^{-\alpha d})n} \leq \left(\frac{en}{2^{-\alpha d}n}\right)^{2^{-\alpha d}n} 2^{-\Omega(\alpha d 2^{-\alpha d})n} \leq 2^{-2^{-O(d)}n}$$

We conclude that *whp* the number of clauses in which all disagreed variables appear is at most $O(d)2^{-\Omega(d)}n = 2^{-\Omega(d)}n$. ∎

Lemma 4.3. *Let F be a planted semi-random 3SAT instance as in Theorem 2. Then whp over the choice of F the number of 0- and 1-clauses w.r.t the approximate assignment found by Alg2 in the first step (Reduction to MAX2SAT) is at most $\frac{n}{O(d\log d)}$ (and the rest are 2- or 3-clauses).*

Proof. Recall that to obtain an approximate assignment for F, $Alg2$ first transforms it to a 2SAT form and then applies a known technique for approximating MAX2SAT.

Take any clause, say, $(x\vee y\vee \sim z)$, and assume it is satisfied by the planted assignment as a 2- or 3-clause. Then for any two literals of this clause, say, x and y, the planted assignment would also satisfy their disjunction $(x\vee y)$, and hence it must satisfy the following 2SAT form as well: $(x\vee y)\wedge(x\vee\sim z)\wedge(y\vee\sim z)$.

Thus if we transform each clause to a 2SAT form in such manner, we may now view the entire formula as a MAX2SAT having an assignment that satisfies a $1-\epsilon$ fraction of the clauses for $\epsilon = \frac{c}{d^4\log^2 d}$ (since we are guaranteed that in the original formula only an ϵ fraction of the clauses are 0- or 1-clauses $w.r.t$ ϕ, and the rest are 2- or 3-clauses). The approximation algorithm for MAX2SAT used in the 'Reduction to MAX2SAT' phase of $Alg2$ satisfies a $1-O(\sqrt{\epsilon})$ fraction of all clauses

([9]). In our case applying such algorithm would result in an assignment *w.r.t* which the number of 0- and 1-clauses is at most $O(\sqrt{\epsilon})m = O(\sqrt{d^{-4}\log^{-2} d})dn = \frac{n}{O(d\log d)}$. ■

4.2 Some technical lemmas

Lemma 4.4. *There exists a constant c such that for every sufficiently large d the following holds: let F be a 3SAT formula with dn clauses and assume the variables of each clause were chosen independently at random. Then whp over the choice of F every subset of i clauses for $i \leq \frac{cn}{d}$ contains at least i distinct variables.*

Proof. We would like to determine the largest possible m for which *whp* over the choice of F every set of i clauses for $i \leq m$ contains at least i distinct variables. We would bound the complement by considering an event with even greater probability, that is, the existence of a set of such size which contains *at most* i distinct variables. Formally we ask when does the following expression converge to 0 as n tends to ∞:

$$S_n = \sum_{i=4}^{m} \binom{dn}{i} \binom{3i}{i} \left(\frac{i}{n}\right)^{2i}$$

For every i we choose the i clauses out of the possible dn, fix i variable positions (these are the distinct candidates) and require variables in all other positions are chosen out of these candidates. We start the summation from 4 since every set of 1, 2 or 3 clauses contains at least 3 distinct variables (no repeated variables within a clause).

Note that here we assume the simpler to analyse model, in which for every clause its variables are chosen independently one of another (this process will result *whp* in at most $O(d)$ invalid clauses with two identical variables each, which would be excluded from the formula).

We upper bound this expression using $\binom{n}{k} \leq (\frac{en}{k})^k$ to obtain:

$$S_n \leq \sum_{i=4}^{m} a_i = \sum_{i=4}^{m} \left(\frac{edn}{i}\right)^i \left(\frac{3ei}{i}\right)^i \left(\frac{i}{n}\right)^{2i} = \sum_{i=4}^{m} (3e^2 d)^i \left(\frac{i}{n}\right)^i$$

We notice that

$$\frac{a_{i+1}}{a_i} = \frac{(3e^2d)^{i+1}\left(\frac{i+1}{n}\right)^{i+1}}{(3e^2d)^i\left(\frac{i}{n}\right)^i} = 3e^2d\left(1+\frac{1}{i}\right)^i\left(\frac{i+1}{n}\right) \leq 3e^2d\cdot e^1\left(\frac{i+1}{n}\right).$$

Taking $m = \frac{n}{O(d)}$ with an appropriately chosen leading constant in $O(d)$ ($6e^3$ should be enough), we guarantee that $\frac{a_{i+1}}{a_i} \leq q$ for some $q < 1$.

Hence this sum may be bounded by an infinite geometric one, as follows:

$$S_n \leq \frac{a_4}{1-q} = \frac{(3e^2d)^4(\frac{4}{n})^4}{1-q} = \frac{(12e^2d)^4}{1-q}\frac{1}{n^4} = O\left(\frac{1}{poly(n)}\right) = o(1).$$

We conclude that every subset of i clauses for $i \leq \frac{n}{O(d)}$ contains at least i distinct variables *whp*. ■

Lemma 4.5. *There exists a constant $c > 0$ ($c = 12$ suffices) for which the following holds: let F be a planted random 3SAT instance as in Theorem 3 and consider the formula graph associated with F. Then whp over the choice of F every vertex induced subgraph of size as small as $\frac{n}{O(d_2)}$ has an average degree of at most c.*

Proof. Consider a set of variables S which induces a subgraph with average degree of at least $c = 6c'$, then this subgraph must contain at least $3c'|S|$ edges and hence there are k clauses containing at least two variables from S each, for $k \in [c'|S|, 3c'|S|]$ (each clause corresponds to either 1 or 3 edges of the graph).

The total number of clauses containing at least two variables from S which are also satisfied by the planted assignment is $l = 7n\binom{|S|}{2}$, so for every k the probability at least k such clauses are actually included in F is at most $\binom{l}{k}(\max\{p_1,p_2,p_3\})^k \leq \binom{l}{k}(d_2n^{-2})^k$. We take the union bound over all possible values of k and over all sets S of size up to $O(d_2^{-2})n$ to estimate the probability of any such set to exist.

$$S_n = \sum_{i=6c'}^{O(d_2^{-2})n}\binom{n}{i}\sum_{k=c'i}^{3c'i} b_k = \sum_{i=6c'}^{O(d_2^{-2})n}\binom{n}{i}\sum_{k=c'i}^{3c'i}\binom{7n\binom{i}{2}}{k}\left(\frac{d_2}{n^2}\right)^k$$

$$
\begin{aligned}
\sum_{k=c'i}^{3c'i} b_k &= \sum_{k=c'i}^{3c'i} \binom{7n\binom{i}{2}}{k} \left(\frac{d_2}{n^2}\right)^k \\
&\leq \sum_{k=c'i}^{3c'i} \left(\frac{7eni^2}{2k}\right)^k \left(\frac{d_2}{n^2}\right)^k \leq \\
&\leq \sum_{k=c'i}^{3c'i} \left(\frac{7eni^2}{2c'i}\right)^k \left(\frac{d_2}{n^2}\right)^k = \\
&= \sum_{k=c'i}^{3c'i} \left(\frac{7ed_2}{2c'}\frac{i}{n}\right)^k \leq \\
&\leq \sum_{k=c'i}^{\infty} \left(\frac{7ed_2}{2c'}\frac{i}{n}\right)^k = \\
&= \left(\frac{7ed_2}{2c'}\frac{i}{n}\right)^{c'i} \left(1 - \frac{7ed_2}{2c'}\frac{i}{n}\right)^{-1} \leq \\
&\leq \left(\frac{7ed_2}{2c'}\frac{i}{n}\right)^{c'i} \left(1 - \frac{1}{2}\right)^{-1} = \\
&= 2\left(\frac{7ed_2}{2c'}\frac{i}{n}\right)^{c'i}
\end{aligned}
$$

where the last inequality is justified by $i \leq O(d_2^{-2})n$ with appropriately chosen constant.

Thus we obtain:

$$
\begin{aligned}
S_n \leq 2 \sum_{i=6c'}^{O(d_2^{-2})n} a_i &= 2 \sum_{i=6c'}^{O(d_2^{-2})n} \left(\frac{en}{i}\right)^i \left(\frac{7ed_2}{2c'}\frac{i}{n}\right)^{c'i} \\
&= 2 \sum_{i=6c'}^{O(d_2^{-2})n} \left(e^{c'+1} \left(\frac{7d_2}{2c'}\right)^{c'} \left(\frac{i}{n}\right)^{c'-1}\right)^i
\end{aligned}
$$

Next we would bound this sum by an infinite geometric one.

$$\frac{a_{i+1}}{a_i} = e^{c'+1}\left(\frac{7d_2}{2c'}\right)^{c'} \frac{\left(\frac{i+1}{n}\right)^{(c'-1)(i+1)}}{\left(\frac{i}{n}\right)^{(c'-1)(i)}} =$$
$$= O(d_2^{c'})\left(1+\frac{1}{i}\right)^{i(c'-1)}\left(\frac{i+1}{n}\right)^{c'-1} \leq$$
$$\leq O(d_2^{c'})e^{c'-1}\left(\frac{i+1}{n}\right)^{c'-1} =$$
$$= O(d_2^{c'})\left(\frac{i+1}{n}\right)^{c'-1}$$

For $c' \geq 2$ (when the average degree $c = 6c'$ is at least 12) and by choosing the appropriate constant in $i \leq O(d_2^{-2})n$ (a constant of $3.5^{-2}e^{-4}$ would be sufficient) we may bound $O(d_2^{c'})\left(\frac{i+1}{n}\right)^{c'-1}$ by some $q < 1$, implying that

$$S_n \leq \frac{2a_{6c'}}{1-q} = \frac{c''}{poly(n)} = o(1)$$

■

Lemma 4.6. *There exists a constant $c > 0$ for which the following holds: let F be a planted random 3SAT instance as in Theorem 3 and consider the formula graph associated with F. Then whp over the choice of F every vertex induced subgraph of size as small as $\frac{n}{O(d_1)}$, where only edges corresponding to 1-clauses w.r.t the planted assignment are considered, has an average degree of at most c.*

Proof. The proof is identical to that of Lemma 4.5 but considers sets of at most $O(d_1^{-1})n$ variables and replaces $\max\{p_1, p_2, p_3\}$ by $p_1 = d_1 n^{-2}$. ■

4.3 Unassignment

Lemma 4.7. *Let F be a 3SAT formula with dn clauses and assume the variables of each clause were chosen independently at random. Consider an arbitrary assignment ψ w.r.t which the number of 0- and 1-clauses is at most $\frac{n}{O(d \log d)}$. Then*

whp over the choice of F a 0-Clause Unassignment which begins with ψ results in at most $\frac{n}{O(d)}$ unassigned clauses.

Proof. Let σ be the final partial assignment of the '0-Clause Unassignment' phase (it is partial to ψ, the initial assignment of that phase). In the following analysis an i-clause is such $w.r.t$ the initial assignment ψ whereas an unassigned clause or variable is such $w.r.t$ the final partial assignment σ.

Let m_{init} be the number of 0- and 1-clauses, and m_{end} be the number of unassigned clauses at the end of the unassignment. Our objective is to upper bound m_{end} by a function of m_{init}. Let $m_{\text{end}} = m_{01} + m_{23}$, where m_{01} is the number of unassigned clauses which are 0- or 1-clauses ($w.r.t$ ψ) and similarly m_{23} is the number of unassigned clauses which are 2- or 3-clauses ($w.r.t$ ψ). The number of distinct variables that appear in the unassigned clauses is at most $3m_{01}+m_{23}$, since each 0- or 1-clause contributes at most three distinct variables, whereas each 2- or 3-clause, in the moment it becomes unsatisfied (and as a result unassigned), must contain at least two variables already unassigned (otherwise it could not become unsatisfied), thus contributing at most one new variable.

We assume towards a contradiction that $m_{\text{end}} \geq cm_{01}$ for some constant c. For the sake of analysis, let us terminate the process at the very iteration when $m_{\text{end}} = cm_{01}$. Also assume for simplicity all 1-clauses are unassigned, that is, $m_{01} = m_{\text{init}}$. Plugging in the parameters according to our assumption we obtain that the number of distinct variables is at most $3m_{01}+m_{23} = 3m_{01}+(m_{\text{end}}-m_{01}) = 3m_{01}+(cm_{01}-m_{01}) = (c+2)m_{01}$; in other words, the maximal possible expansion of the set of unassigned clauses is $1+\frac{2}{c}$. Our strategy is to show that for small enough m_{init} every set of size at most cm_{init} is *whp* at least $(1+\frac{2}{c})$-expanding, which would imply that m_{end} must be in fact smaller than cm_{init}. To be more concrete, we are looking for the largest m_{end} such that every set of at most m_{end} clauses is at least $(1+\frac{2}{c})$-expanding *whp*. Similarly to Lemma 4.4, we ask what is the maximal m_{end} for which the following sum converges to 0 when n tends to ∞:

$$S_n = \sum_{i=3}^{m_{\text{end}}} \binom{dn}{i} \binom{3i}{(1+\frac{2}{c})i} \left(\frac{(1+\frac{2}{c})i}{n}\right)^{[3-(1+\frac{2}{c})]i}$$

For every i we choose the i clauses out of the possible dn, fix $(1+\frac{2}{c})i$ variable

positions, which serve as the distinct candidates, and require all variables in other positions are chosen out of these candidates.
Next, we upper bound this expression using $\binom{n}{k} \leq (\frac{en}{k})^k$ to obtain:

$$S_n \leq \sum_{i=3}^{m_{\text{end}}} a_i = \sum_{i=3}^{m_{\text{end}}} \left(\frac{edn}{i}\right)^i \left(\frac{3ei}{(1+\frac{2}{c})i}\right)^{(1+\frac{2}{c})i} \left(\frac{(1+\frac{2}{c})i}{n}\right)^{(2-\frac{2}{c})i} =$$
$$= \sum_{i=3}^{m_{\text{end}}} \left(3^{1+\frac{2}{c}} \left(1+\frac{2}{c}\right)^{1-\frac{4}{c}} e^{2+\frac{2}{c}} d\right)^i \left(\frac{i}{n}\right)^{(1-\frac{2}{c})i};$$

Hence,

$$\frac{a_{i+1}}{a_i} = \frac{\left(3^{1+\frac{2}{c}} \left(1+\frac{2}{c}\right)^{1-\frac{4}{c}} e^{2+\frac{2}{c}} d\right)^{i+1} \left(\frac{i+1}{n}\right)^{(1-\frac{2}{c})(i+1)}}{\left(3^{1+\frac{2}{c}} \left(1+\frac{2}{c}\right)^{1-\frac{4}{c}} e^{2+\frac{2}{c}} d\right)^i \left(\frac{i}{n}\right)^{(1-\frac{2}{c})i}} =$$
$$= \left(3^{1+\frac{2}{c}} \left(1+\frac{2}{c}\right)^{1-\frac{4}{c}} e^{2+\frac{2}{c}} d\right) \left(1+\frac{1}{i}\right)^{i(1-\frac{2}{c})} \left(\frac{i+1}{n}\right)^{1-\frac{2}{c}} \leq$$
$$\leq \left(3^{1+\frac{2}{c}} \left(1+\frac{2}{c}\right)^{1-\frac{4}{c}} e^{2+\frac{2}{c}} d\right) e^{1-\frac{2}{c}} \left(\frac{i+1}{n}\right)^{1-\frac{2}{c}} =$$
$$= \left(3^{1+\frac{2}{c}} \left(1+\frac{2}{c}\right)^{1-\frac{4}{c}} e^3 d\right) \left(\frac{i+1}{n}\right)^{1-\frac{2}{c}}.$$

This time we need to choose $m_{\text{end}} < \dfrac{n}{\left(3^{1+\frac{2}{c}} \left(1+\frac{2}{c}\right)^{1-\frac{4}{c}} e^3 d\right)^{\frac{c}{c-2}}}$ to have $\dfrac{a_{i+1}}{a_i} \leq q$

for some $q < 1$, so we can bound S_n by the following infinite geometric sum:

$$\begin{aligned} S_n \leq \frac{a_3}{1-q} &= \frac{\left(3^{1+\frac{2}{c}}\left(1+\frac{2}{c}\right)^{1-\frac{4}{c}} e^{2+\frac{2}{c}} d\right)^3}{1-q}\left(\frac{3}{n}\right)^{(1-\frac{2}{c})3} = \\ &= \frac{\left(3^2\left(1+\frac{2}{c}\right)^{1-\frac{4}{c}} e^{2+\frac{2}{c}} d\right)^3}{1-q}\frac{1}{n^{(1-\frac{2}{c})3}} = \\ &= O\left(\frac{1}{poly(n)}\right) \\ &= o(1). \end{aligned}$$

To complete the proof, we require that $m_{\text{end}} \leq \ cm_{\text{init}} \leq \frac{n}{O(d)}$.

Considering the maximal m_{end} possible, that is, $\frac{n}{O(d^{\frac{c}{c-2}})}$, we obtain the constraint:

$$\frac{c}{d^{\frac{c}{c-2}}} \leq \frac{1}{d},$$

or,

$$c \leq d^{\frac{2}{c-2}}$$

which implies

$$(c-2)\log c \leq 2\log d.$$

Taking $c = O\left(\frac{\log d}{\log\log d}\right)$ we guarantee the above constraint is met (by simple algebraic manipulations).

We conclude that when $m_{\text{init}} \leq \frac{n}{O(d\log d)}$ then $m_{\text{end}} \leq \frac{n}{O(d)}$. ■

Lemma 4.8. *Let F be a planted random 3SAT instance as in Theorem 3. Then whp over the choice of F, at the end of the 'Small-Support Unassignment' phase of Alg3 there are at least $(1-2^{-\Omega(d_1)})n$ assigned variables and the assignment of all assigned variables agrees with the planted assignment ϕ.*

Proof. The proof consists of two parts. First we identify a set of variables of size $(1-2^{-\Omega(d_1)})n$ on which the planted assignment ϕ and the Majority Vote maj agree and which remains assigned during the unassignment phase. Secondly we show no variables on which ϕ and maj disagree 'survive' the unassignment phase.

For the sake of analysis of the first part, consider exactly the same unassignment process as described in $Alg3$ in which prior to the unassignment all variables on which ϕ and maj disagree are unassigned as well (denote this set by A). Clearly, such a process can result in only fewer assigned variables than the original process, and therefore it is enough to identify a set as described above for the new process. Let also B be the set of variables which support less than $d_1/2$ clauses $w.r.t$ maj and let C be the set of variables removed during iterations. Let S be the set of variables which remain assigned at the end of the new defined process, $S = \overline{A \cup B \cup C}$. Lemma 4.1 guarantees that $|A| \leq 2^{-\Omega(d_2)}n$. Also, $|B| \leq 2^{-\Omega(d_1)}n$ by the following argument: it is enough to consider both variables which support less than $d_1/2$ clauses $w.r.t$ ϕ and those which appear in some clause together with a variable on which maj and ϕ disagree. Any individual variable x supports a total number of n^2 clauses and each such clauses is actually included in the formula with probability $p_1 = d_1/n^2$, so x is expected to support exactly d_1 clauses ($w.r.t$ ϕ). The number of clauses x supports is a binomial random variable and strongly concentrated around its mean; hence x has probability $2^{-\Omega(d_1)}$ of supporting less than $d_1/2$ clauses. The expected number of variables which support less than $d_1/2$ clauses each is therefore $2^{-\Omega(d_1)}n$ and by considering the martingale of the clause selection process the actual number is indeed $2^{-\Omega(d_1)}n$ whp (similarly to Lemma 4.1). In addition, as explained in Lemma 4.2, the number of clauses which contain some variable on which maj and ϕ disagree is whp at most $2^{-\Omega(d_2)}n$. The total number of variables that appear in such number of clauses is at most $3 \cdot 2^{-\Omega(d_2)}n = 2^{-\Omega(d_2)}n$. To summarize, whp there are at most $2^{-\Omega(d_1)}n + 2^{-\Omega(d_2)}n \leq 2^{-\Omega(d_1)}n$ variables which support less than $d_1/2$ clauses each.

Assume towards a contradiction that at some iteration C has reached the size of $|A|+|B|$ and consider the formula graph induced by the variables of A, B and C on that iteration. By our assumption it has $2(|A| + |B|) \leq 2(2^{-\Omega(d_2)} + 2^{-\Omega(d_1)})n \leq 2^{-\Omega(d_1)}n < O(d_1^{-2})n$ vertices. On the other hand, the average degree depends on d_1 since each variable supports ($w.r.t$ maj) at least $(1/2 - 1/3)d_1 = d_1/6$ clauses with variables from $A \cup B \cup C$. All these edges correspond to 1-clauses also $w.r.t$ ϕ, except for maybe $2^{-\Omega(d_2)}n$ 2- and 3-clauses $w.r.t$ ϕ which are 1-clauses $w.r.t$ maj, as mentioned above. The total contribution of these clauses is negligible since their number is much smaller than the number of the variables involved and

hence we can ignore them. Therefore, the average degree is still $\Omega(d_1)$ with all edges corresponding to 1-clauses, contradicting Lemma 4.6. We conclude that $|C| \leq |A| + |B| \leq 2^{-\Omega(d_1)}n$ which implies that $|S| \geq n - 2|C| \geq (1 - 2^{-\Omega(d_1)})n$.

For the second part, by Lemma 4.1 *whp* there are at most $2^{-\Omega(d_2)}n$ variables on which ϕ and maj disagree. Consider the set O consisting of such variables that have also survived the unassignment. Then each variable of O must support at least $d_1/3$ fully assigned clauses, otherwise it would have been unassigned during the iterative process of the unassignment (the support is *w.r.t* σ, the final assignment of the unassignment phase; since it is partial to maj and these clauses are fully assigned by σ the support is *w.r.t* maj as well). In each such clause there is another variable of O since it is a 1-clause *w.r.t* maj and the satisfying variable is assigned differently than in ϕ. Consider the formula graph induced by the variables of O. It is of size at most $2^{-\Omega(d_2)}n$ with average degree at least $d_1/3$, in contradiction to Lemma 4.5. ■

4.4 Completing the partial assignment

Lemma 4.9. *Let F be a planted random 3SAT instance as in Theorem 3. Then whp over the choice of F, at the end of the 'Small-Support Unassignment' phase of Alg3 the formula graph induced by the unassigned variables has connected components of size at most* $\log n$.

Proof. Our objective is to estimate the probability that the formula graph induced by the set of unassigned variables (during the unassignment phase) contains a connected component of size $\log n$. Towards estimating this, we start from a fixed set T of $\log n$ variables and ask what is the probability that the following two events occurred simultaneously: 1. T has been all unassigned by the un assignment process. 2. The formula graph induced by T is connected, or, if we rephrase it, the formula contains a set of clauses such that the corresponding formula graph induced by T is connected. In fact, it is enough to consider a minimal set of such clauses I' (in this context we say a set of clauses is minimal if deleting any clause disconnects the corresponding subgraph). Two types of clauses are to be considered: 'type 1' clauses, which contain two variables from T, and 'type 2' clauses, which contain three variables from T. We think of selecting a set of clauses that would induce

a connected subgraph on T as the following process: we begin with a subgraph containing the variables of T but no edges and add one clause at a time. Then any 'type 1' clause can reduce the number of connected components by at most 1, whereas 'type 2' reduces it by at most 2 (for a minimal set we obtain exactly 1 and 2, respectively[1]). Let t denote the number of variables in T and t_i denote the number of 'type i' clauses. In the initial state of the subgraph the number of connected components is t and in the final state this number is 1, which gives us the following constraint: $t_1 + 2t_2 = t - 1$. First we analyse the probability of the second event to occur for any set T of size t; denote this probability by P_t.

$$P_t \leq \sum_{T,I'} Pr[I' \subset F] \leq \binom{n}{t} \sum_{t_1+2t_2=t-1} \binom{7nt^2}{t_1} \binom{7t^3}{t_2} \left(\frac{d_2}{n^2}\right)^{t_1+t_2}$$

where $\binom{n}{t}$ is the number of possibilities to choose t variables out of the total n; $\binom{7nt^2}{t_1}$ is the number of possibilities to choose 'type 1' clauses (satisfied by the planted assignment ϕ); $\binom{7t^3}{t_2}$ is the number of possibilities to choose 'type 2' clauses (satisfied by ϕ); and $\left(\frac{d_2}{n^2}\right)^{t_1+t_2}$ is the probability that all chosen clauses are actually included in the formula. The cases $t_1 = 0$ and $t_2 = 0$ are simpler, thus we perform

[1]In fact even in a minimal set there might be 'type 2' clauses reducing the number of connected components only by 1. We may think of such clauses as 'type-1' clauses having their third variable in T and therefore such cases are also treated by our calculation.

the analysis assuming neither t_1 nor t_2 are 0.

$$
\begin{aligned}
P_t &\leq \left(\frac{en}{t}\right)^t \sum_{t_1=1}^{t-3} \left(\frac{7ent^2}{t_1}\right)^{t_1} \left(\frac{7et^3}{t_2}\right)^{t_2} \left(\frac{d_2}{n^2}\right)^{t_1+t_2} = \\
&= \left(\frac{en}{t}\right)^t \sum_{t_1=1}^{t-3} (7ed_2)^{t_1+t_2} \left(\frac{t}{n}\right)^{t_1+2t_2} \left(\frac{t}{t_1}\right)^{t_1} \left(\frac{t}{t_2}\right)^{t_2} \leq \\
&\leq \left(\frac{en}{t}\right)^t \sum_{t_1=1}^{t-3} (7ed_2)^{t} \left(\frac{t}{n}\right)^{t-1} \left(\frac{t}{t_1}\right)^{t_1} \left(\frac{t}{t_2}\right)^{t_2} \leq \\
&\leq n(7e^2d_2)^t \binom{t}{t_1}\binom{t}{t_2} \leq \\
&\leq n(7e^2d_2)^t 2^t 2^t = \\
&= n(28e^2d_2)^t.
\end{aligned}
$$

Back to the first event: let S denote the set of variables which have survived the unassignment, to be consistent with the notation of Lemma 4.8 (and so $\overline{S}$ denotes the set of variables which have been unassigned). From this lemma the probability that all variables of T have been unassigned is $2^{-\Omega(d_1)t}$. Let I' be a set of clauses as before. In general it holds that

$$
\begin{aligned}
&\sum_{T,I'} Pr[(I' \subset F) \cap (T \subset \overline{S})] = \sum_{T,I'} Pr[(I' \subset F)] Pr[(T \subset \overline{S})|(I' \subset F)] \\
&= P_t Pr[(T \subset \overline{S})|(I' \subset F)]
\end{aligned}
$$

If the two events were independent, it would hold that $Pr[(T \subset \overline{S})|(I' \subset F)] \leq Pr[(T \subset \overline{S})]$, hence the desired probability (of both events to occur) would be at most $P_t \cdot Pr[T \subset \overline{S}] \leq n(28e^2d_2)^t \cdot 2^{-\Omega(d_1)t}$, so for $t = \log n$, large enough d_1 and $d_2 \leq 2^{O(d_1)}$, it could be upper-bounded by $2^{-\Omega(\log n)} = o(1)$ which completes the proof. However, this is not the case and our strategy would be to consider a particular $T' \subset T$ and a slightly modified unassignment process resulting in a set S' for which the two events would indeed be independent. Also note that for any $T' \subset T$ it holds that $Pr[(T \subset \overline{S})|(I' \subset F)] \leq Pr[(T' \subset \overline{S})|(I' \subset F)]$, hence it is sufficient to show that for a particular such T' (defined below) it is true that $Pr[(T' \subset \overline{S})|(I' \subset F)] \leq Pr[(T' \subset \overline{S'})]$.

Let again I' be a minimal set of clauses, then the number of variables of T that appear at most 6 times in I' is at least $t/2$ (otherwise there would be at least $\frac{6t}{3\cdot 2} = t$ clauses in I' in contradiction to minimality). Denote this subset of variables of T by T'.

For the sake of analysis, consider an unassignment process as suggested in Lemma 4.8 where A contains not only the variables on which the Majority Vote maj and the planted assignment ϕ disagree, but also those variables on which the two assignments agree but the bias of maj towards their assignment in ϕ is at most 6. In addition, the following two sets would be removed prior to the iterations as well: 1. $T \backslash T'$ (which contains at most $t/2$ variables); 2. every variable not in T which appears more than 6 times in I' (there are at most $t/6$ such variables since I' has at most t clauses and every clause has at most one variable not in T). Denote by S' the set of variables which have survived the modified process. When we would like to emphasize that the survival set S' (or S) is defined $w.r.t$ a particular set of clauses I, we denote it by S'_I (or S_I). In the following analysis I denotes the set of clauses of the formula F.

Claim 1. $S'_I \subset S_{I \cup I'}$

Proof. First, we show that this relation holds prior to the iterations. Let x be a variable in $\overline{S_{I \cup I'}}$ (prior to the iterations), we will show that x is also in $\overline{S'_I}$. We distinguish two cases: if $x \in T \backslash\ T'$ or not in T but appears more than 6 times in I' then it is in $\overline{S'_I}$ by the definition of the new process; if $x \notin T \backslash\ T'$ or not in T and appears at most 6 times in I' then again we have two cases: if x is unassigned because it supports less than $d_1/2$ clauses $w.r.t\ I \cup I'$ then all the more it would support less than $d_1/2$ clauses $w.r.t\ I$. If it is unassigned because maj and ϕ disagree on its assignment, removing the clauses of I' in which it appears at most 6 times can result in a bias of at most 6 towards its assignment in ϕ and hence it is also in $\overline{S'_F}$. We notice that this already implies the assertion of the claim since the iterations are defined equally for both processes. ■

Next, let $F = I$ denote the event: 'I is the set of clauses of the formula F'.

We have:

$$\begin{aligned} Pr[(T' \subset \overline{S'})] &= \sum_{I:T' \subset \overline{S'_I}} Pr[F = I] \geq \\ &\geq \sum_{I:T' \subset \overline{S_{I \cup I'}}} Pr[F = I] \geq \\ &\geq \sum_{I'':I'' \cap I' = \emptyset, T' \subset \overline{S_{I'' \cup I'}}} Pr[F = I'' \cup I'] = \\ &= Pr[(T' \subset \overline{S}) | (I' \subset F)] \end{aligned}$$

where the first inequality is due to the claim.

It remains to bound the probability $Pr[(T' \subset \overline{S'})]$. Indeed, *w.r.t* every choice of S' (which depends on $T \backslash T'$ and $\overline{T}$, i.e., it depends only on $\overline{T'}$) a very similar argument to that of Lemma 4.8 guarantees that S' is of size at least $2^{-\Omega(d_1)}n$, *whp* (we notice that prior to the iterations only additional $O(\log n)$ variables could have been unassigned). Since this choice does not depend on T', we conclude that $Pr[(T' \subset \overline{S'})] \leq 2^{-\Omega(d_1 \log n)}$, which completes the proof. ■

4.5 Proofs of the theorems

Proof of Theorem 1. Lemma 4.2 guarantees that during its approximation phase, $Alg1$ will find an assignment *w.r.t* which the number of 0- and 1-clauses in F is at most $\frac{n}{O(d \log d)}$ (and so the rest are 2- are 3-clauses). Lemma 4.7 implies that in such case the number of unassigned clauses at the end of the '0-Clause Unassignment' phase of $Alg1$ is *whp* at most $\frac{n}{O(d)}$. Next, Lemma 4.4 guarantees that *whp* every subset of this set of clauses is 1-expanding. Now by Hall's condition there is a matching of clauses to variables, which means that $Alg1$ would find a matching in the bipartite graph constructed during its 'Matching' phase (without spoiling the rest of the clauses which are already satisfied by other variables). ■

Proof of Theorem 2. Similar to the proof of Theorem 1, substituting Lemma 4.2 by Lemma 4.3. ■

Proof of Theorem 3. The Majority Vote computed by $Alg3$ during its approxima-

tion phase agrees with the planted assignment on at least $(1-2^{-\Omega(d_2)})n$ variables *whp*, as promised by Lemma 4.1. When the 'Small-Support Unassignment' phase begins with such an assignment, Lemma 4.8 implies that *whp* the partial assignment obtained at the end of that phase agrees with the planted assignment and that at least $(1-2^{-\Omega(d_1)})n$ variables remain assigned. In such case we are guaranteed the residual formula (which is induced by the unassigned variables) is indeed satisfiable. Then by Lemma 4.9 the connected components of this formula's graph are *whp* of size at most $\log n$, which guarantees that $Alg3$ would complete the assignment successfully in polynomial time during its exhaustive search phase. ■

References

[1] A. Agarwal, M. Charikar, K. Makarychev, Y. Makarychev. $O(\sqrt{logn})$ approximation algorithms for Min UnCut, Min 2CNF Deletion, and directed cut problems. STOC, 2005.

[2] N. Alon and U. Feige. On the power of two, three and four probes. SODA, 346-354, 2009.

[3] N. Alon and N. Kahale. A spectral techniques for coloring random 3-colorable graphs. SIAM Journal of computation 26(6), 1733-1748, 1997.

[4] N. Alon, M. Krivelevich and B. Sudakov, Finding a large hidden clique in a random graph, Random Structures and Algorithms 13, 457-466, 1998.

[5] E. Ben-Sasson, Y. Bilu and D. Gutfreund. Finding a Randomly Planted Assignment in a Random 3-CNF. Manuscript, 2002.

[6] A. Blum, and J. Spencer. Coloring random and semirandom k-colorable graphs. Journal of Algorithms 19, 204-234, 1995.

[7] R. B. Boppana, Eigenvalues and graph bisection: an average case analysis. FOCS, 280-285, 1987.

[8] A. Braunstein, M. Mezard, and R. Zecchina, Survey propagation: an algorithm for satisfiability. Random Structures Algorithms 27(2), 201-226, 2005.

[9] M. Charikar, K. Makarychev, Y. Makarychev. Near-optimal algorithms for maximum constraint satisfaction problems. SODA, 2007.

[10] A. Coja-Oghlan, M. Krivelevich, and D. Vilenchik. Why almost all satisfiable k-CNF formulas are easy. In proc. 13th International Conference on Analysis of Algorithms, DMTCS proc., 89-102, 2007.

[11] S. Cook. The complexity of theorem-proving procedures. FOCS, 151-158, 1971.

[12] J.M. Crawford and L.D. Auton. Experimental Results on the Crossover Point in Random 3SAT. Artificial Intelligence, 81, 1996.

[13] U. Feige, and J. Kilian. Heuristics for Semirandom Graph Problems. Journal of Computer and System Sciences 63, 639-671, 2001.

[14] U. Feige and R. Krauthgamer. Finding and certifying a large hidden clique in a semi random graph. Random Structures and Algorithms 16(2), 195-208, 2000.

[15] U. Feige and D. Vilenchik. A Local Search Algorithm for 3SAT. Technical Report MCS 04-07, Computer Science and Applied Mathematics, The Weizmann Institute of Science, 2004.

[16] A. Flaxman. A spectral technique for random satisfiable 3CNF formulas. SODA, 357-363, 2003.

[17] A. Flaxman. Algorithms for Random 3-SAT, extended version of chapter in Encyclopedia of Algorithms, 742-744, 2008.

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

[19] A. C. Kaporis, L. M. Kirousis. E. G. Lalas, The probabilistic analysis of a greedy satisfiability algorithm. Random Structures and Algorithms, Wiley.

[20] A. C. Kaporis, L. M. Kirousis, Y. C. Stamatiou, M. Vamvakari, and Michele Zito. Coupon collectors, q-binomial coefficients and the unsatisfiability threshold. ICTCS, 328-338, 2001.

[21] E. Koutsoupias, and C. H. Papadimitriou. On the greedy algorithm for satisfiability. Inform. Process. Lett. 43 (1), 53-55, 1992.

[22] L. A. Levin. Universal enumeration problems. Problemy Peredachi Informacii 9(3), 115-116, 1973.

[23] M. Molloy, Cores in random hypergraphs and Boolean formulas. Random Structures Algorithms 27(1), 124-135, 2005.

[24] D. Vilenchik. Finding a satisfying assignment for semi-random satisfiable 3CNF formulas. Master Thesis, The Weizmann Institute of Science, 2004.