# Selecting Complementary Pairs of Literals*

Alexis C. Kaporis, Lefteris M. Kirousis, and Efthimios G. Lalas

University of Patras, Department of Computer Engineering and Informatics
University Campus, GR-265 04, Patras, Greece
{kaporis,kirousis,lalas}@ceid.upatras.gr

**Abstract.** We present and rigorously analyze a heuristic that searches for a satisfying truth assignment of a given random instance of the 3-SAT problem. We prove that the heuristic asymptotically certainly succeeds in producing a satisfying truth assignment for formulas with clauses to variables ratio (density) of up to 3.52. Thus the experimentally observed threshold of the density where a typical formula's phase changes from asymptotically certainly satisfiable to asymptotically certainly contradictory is rigorously shown to be at least 3.52. The best previous lower bound in the long series of mathematically rigorous approximations by various research groups of the experimental threshold was 3.42. That was the first result where the probabilistic analysis was based on random formulas with a pre-specified, typical number of appearances for each literal. However, in that result, in order to simplify the analysis, the number of appearances of each literal was decoupled from the number of appearances of its negation. In this work, we assume not only that each literal has the typical number of occurrences, but that for each *variable* both numbers of occurrences of its positive and negated appearances are typical. By standard techniques, our algorithm can be easily modified to run in linear time. Thus not only the satisfiability threshold, but also the threshold (experimental again) where the complexity of searching for satisfying truth assignments jumps from polynomial to exponential is at least 3.52. This should be contrasted with the value 3.9 for the complexity threshold given by theoretical (but not mathematically rigorous) techniques of Statistical Physics.

## 1 Introduction

Consider $n$ Boolean variables $V = \{x_1, \dots, x_n\}$, where each $x_i$ may appear as positive literal $x_i$ or as negative $\overline{x}_i$ in a $k$-clause. The set of $2n$ possible literals is $L = \{x_1, \overline{x}_1, \dots, x_n, \overline{x}_n\}$. A $k$-clause is a disjunction of $k$ literals of distinct underlying variables. A random $k$-CNF formula is the conjunction of $m$ clauses, each selected amongst $2^k\binom{n}{k}$ possible, uniformly and independently (standard $G_{n,m}$ model). A formula $\phi$ is satisfiable if each variable $x_i \in V$ is assigned a value in $\{0, 1\}$ such that $\phi$ evaluates to 1. It is conjectured that for all $k \geq 2$ there exists a critical clauses to variables $m/n$ constant ratio $c_k^*$ such that almost

* Research supported by the University of Patras Research Committee Project Carathéodory under contract no. 2445

all formulas with ratio (greater)-less than $c_k^*$ are (un)-satisfiable, [30]. For $k = 2$ it has been proven that $c_2^* = 1$, see [8, 15, 19]. Friedgut [16], proved the existence of a sequence of threshold values $c_k(n)$ for $k$-CNF formulas. The convergence $\lim_{n\to\infty} c_k(n) = c_k^*$ still remains open. In a recent advance Frieze and Wormald [18] proved that $c_k^* \sim 2^k \ln 2$ as $k - \ln_2 n \to \infty$. Independently, Achlioptas and Moore [3] proved the general lower bound $c_k^* \geq \frac{2^k}{2} \ln 2 - z$ ($z$ is a constant) for any *fixed* value of $k \geq 2$.

For $k = 3$, upper bounds to $c_3^*$ have been proven using probabilistic counting arguments, [11, 12, 21, 22, 25–27]. The current best upper bound is $c_3^* < 4.506$ by Dubois, Boufkhad and Mandler in [12].

On the other side, almost all lower bounds are algorithmic. To prove that $c_3^* > c$ it suffices to prove, as a consequence of [16], that on input a random formula of density $c$ a Davis-Putnam algorithm computes a satisfying truth assignment with probability of at least a positive constant, see [1, 4–8, 17, 23]. The current best lower bound is $c_3^* > 3.42$ proved by Kaporis Kirousis and Lalas in [23].

## 2 Motivation and Contribution

This work was motivated from [23] where a heuristic that in each free step Selects & Sets to TRUE a literal as to satisfy the maximum number of clauses was analyzed. Despite it's simplicity this algorithm gave a substantial improvement over the existing lower bounds. It proposes the following ordering amongst unset literals: give highest priority for satisfaction to literals of the highest degree. The degree $deg(\tau)$ of a literal $\tau \in L$ is the number of its occurrences in the formula, irrespective of clause sizes. It's analysis was greatly simplified by the fact that it selects a literal $\tau \in L$ of the highest current degree, irrespective of $deg(\overline{\tau})$.

Consider literals $\tau_1, \ldots, \tau_s \in L$, all of the highest degree in the current formula. Then it seems natural to give priority for satisfaction to literal $\tau_i, i \in \{1, \ldots, s\}$, that it's negation $\overline{\tau}_i$ un-satisfies the fewest clauses. Intuitively, obtaining control on *complementary* literals $\tau, \overline{\tau} \in L$ we maximize the number of satisfied clauses and minimize the generation of new 1-clauses increasing the probability of success of the algorithm. However, the ordering of selection of such pairs of literals is not trivial. For example: let $\tau_1$ with $deg(\tau_1) = \Delta$ be the unique literal of currently maximum degree and $deg(\overline{\tau_1}) = 3$. Should we better select literal $\tau_2$ with $deg(\tau_2) = \Delta - 1$ and $deg(\overline{\tau_2}) = 2$? Here, each pair of complementary literals has *discrepancy* $\Delta - 3$.

We provide a framework for analyzing algorithms that under an *arbitrary rule* Select & Set a pair of complementary literals per free step, irrespectively of the clause sizes. The amount of information that we have to keep track of per step is increased significantly to a *double* degree sequence in contrast to the *single* one used in [23]. A Theorem proved by Wormald in [33] is applied to approximate the dynamics of this sequence with high probability.

We analyze algorithm `CL`. Intuitively, in each free step Selects & Sets a pair of literals such as to minimize the expected number of forced steps (steps that we satisfy all subsequent 1-clauses) per round. It succeeds with positive probability for densities $c < 3.52+$.

**Theorem 1.** *The conjectured threshold value for random 3-CNF formulas is* $c_3^* > 3.52+$.

In a series of papers [13, 14, 28, 29] were evaluated, employing methods of statistical physics, some heuristics with remarkable performance. The operation of these heuristics depends heavily in exploiting degree information over the literals of the formula. We expect that the tools developed in this paper will be useful for analyzing rigorously such heuristics. Hajiaghayi and Sorkin independently analyzed heuristics similar to those of this paper and claim to have obtained the same lower bound [20].

## 3 Algorithm

Let $h$ be a natural number (here we set $h = 10$).

Definition 1

$$\mathcal{X}_{i,j} = \{\tau \in L \mid deg(\tau) \succeq_h i \text{ and } deg(\overline{\tau}) \succeq_h j\},\ (i,j) \in A = \{0,\ldots,h\}^2,$$

*where we define the relation* $\succeq_h$ *as:*

$$deg(\tau) \succeq_h i \Leftrightarrow \begin{cases} deg(\tau) = i, & i < h, \\ deg(\tau) \geq h, & i = h. \end{cases}$$

*If a literal* $\tau \in L$ *belongs in* $\mathcal{X}_{h,i}, i = 0,\ldots,h$ *is called heavy, otherwise is called light.*

*Remark 1.* If $\tau \in \mathcal{X}_{i,j}$ then $\overline{\tau} \in \mathcal{X}_{j,i}$ and $|\mathcal{X}_{i,j}| = |\mathcal{X}_{j,i}|$, $\forall\ (i,j) \in A$. Also, if $\tau \in \mathcal{X}_{i,i}$ then $\overline{\tau} \in \mathcal{X}_{i,i},\ 0 \leq i \leq h$.

`Del&Shrink`($\tau$) is the Davis-Putnam rule: delete any clause that contains literal $\tau$ and delete $\neg\tau$ from any clause it appears.

```
Algorithm: CL
begin:
  while unset literals exist do:
    (s,t)← Choose-Bucket;
        Select τ ∈ X_{s,t} & Set τ = 1;
        Del&Shrink(τ);
        while 1-clauses exist do:
          Select τ in a 1-clause & Set τ = 1;
          Del&Shrink(τ);
        end do;
  end do;
end;
```

`Procedure Choose-Bucket:` Compute $m_2, m_3$ of the current formula $\phi$. For each $\mathcal{X}_{i,j} \neq \emptyset$, select at random $\tau \in \mathcal{X}_{i,j}$ and perform 1 round. Compute the new $m_2', m_3'$. Compute $R(i,j) = \frac{m_2'-m_2}{m_3'-m_3}$. Return to the initial $\phi$ and repeat. Store the pair of indices $(i,j) \in A$ that maximized $R(i,j)$.

For more details about the procedure `Choose-Bucket` see Section 10. The rest of this paper is organized in the following Sections. In Section 4 we define the notion of a *round* of algorithm's operation. Also we point out why we use the current number of rounds as the "time" parameter. In Section 5, we describe a balls in bins model of generating the random formula obtained at the end of each round of the algorithm. In Section 6 we prove various statistical properties of the collections of available literals per round. In Section 7 we prove that the length of each round is $o(n)$ with high probability. Also we prove that the probability of success is bounded away from zero. In Section 8 we establish a system of differential equations describing the dynamics of the algorithm per round and, we prove that the conditions of Wormald's theorem hold in Section 9. Finally in Section 10 we implement the solution of the differential equations and apply a criterion of termination of the algorithm.

## 4 Rounds

A *round* is an iteration of the first `while` loop of the algorithm `CL`. It consists of a single free atomic step and a sequence of forced atomic steps in which all 1-clauses are satisfied (inner `while` loop). At the end of each round only 3, 2-clauses exist, and we reach the same simplified formula independently of the ordering that `CL` satisfies the 1-clauses. Therefore, applying Theorems 1, 2 in [33], we trace the cardinalities of 3, 2-clauses per round, ignoring the 1-clauses that may appear *into* the round. The idea of using the number of rounds as time parameter was introduced in [4]. We can apply Wormald's Theorem to approximate w.h.p. any random variable with the following concentration property: Knowledge of it's value only within $o(n)$, suffices to determine it's expected change within $o(1)$. Exactly this is the problem of tracing the number of 1-clauses per atomic step: the expected change of the number of 1-clauses, when there are some 1-clauses present, is discontinuously different when there is none.

## 5 Balls in Bins

The random process `Balls-in-Bins` (2) that we use to construct a random formula given some parameters is a generalization of the model also used in [5, 12, 31, 18, 3]. Each formula obtained from this process has $L$ as it's set of literals; $C_i, i = 2, 3$ as it's current set of $i$-clauses and $\mathcal{X}_{i,j}, i, j \in \{0, \ldots, h\}^2$, as it's set of complementary literals according to Definition (1). Let $n$ the number of variables of the initial formula (before the algorithm starts) and a predetermined parameter $h \in N$. We define the following $(h+1)^2+3$ scaled with respect to $n$ parameters: $\ell = |L|/n$, the scaled number of unset literals. $c_i = |C_i|/n, i = 3, 2$, the scaled number of $i$-clauses. According to Definition (1), $x_{i,j} = |\mathcal{X}_{i,j}\ |/n$, the scaled number of corresponding complementary literals, $0 \leq i, j \leq h$. Notice $\sum_{i,j}^{h} x_{i,j} = \ell$ and the summation over all literals in $L$ of their corresponding degrees equals the total number of literal occurrences $pn = 3(c_3 + 2c_2)n$. Using the random process (2) we construct a random formula given the above $(h+1)^2+3$ scaled parameters, namely given the vector:

$$\mathcal{S} = \langle \ell, c_3, c_2, x_{0,0}, x_{0,1}, \ldots x_{0,h}, \ldots, x_{h,0}, x_{h,1}, \ldots, x_{h,h} \rangle, \tag{1}$$

Now, we define the notion of a ball, a bin and an (un)exposed register. Each one position in a clause is a *ball* and thus we obtain a total of $pn = (3c_3 + 2c_2)n$ distinct balls. Each ball$_i$, is adjacent with 3 registers, *clause-ball$_i$*, *neighbor-ball$_i$* and *negation-ball$_i$*, $i = 1, \ldots, pn$. Each literal in $L$ is a *bin* and so we have $\ell n$ distinct bins. Furthermore, each bin$_j$ is adjacent with registers *load-bin$_j$* and *negation-bin$_j$*, $j = 1, \ldots, \ell n$.

Definition 2 *A register is exposed if we know it's current content and we make use of it to decide the next algorithmic step, otherwise the register is unexposed.*

Definition 3 *Exposed register clause-ball$_i$ contains integer $k$ iff ball$_i$ appears in a $k$-clause, $k \in \{0,1,2,3\}$. Unexposed register neighbor-ball$_i$ contains $k-1$ pointers pointing to the $k-1$ distinct balls sharing the same $k$-clause with ball$_i$, $k \in \{3,2\}$. Unexposed register negation-ball$_i$ points to the bin that represents the negation of the underlying literal of ball$_i$. For each bin$_j$, the exposed register load-bin$_j$ contains the pair of integers $(load_s^h, load_t^h)$ iff the actual load of bin$_j$ is $s$ and the actual load of the bin that corresponds to the negation of the literal of bin$_j$ is $t$. Unexposed register negation-bin$_j$ points to the bin that represents the negation of the literal of bin$_j$. Also:*

$$load_s^h = \begin{cases} s, & s < h, \\ h, & s \geq h. \end{cases}$$

The process that follows constructs a random formula given the vector $\mathcal{S}$ in (1).

$$\texttt{Balls-in-Bins} \tag{2}$$

1. Select randomly $2c_2n$ out of $pn = (3c_3+2c_2)n$ distinct balls and set to 2 their clause-ball register, similarly set to 3 the registers of the remaining balls.
2. Construct randomly $c_in$ distinct $i$-tuples from the $ic_in$ balls, $i = 2, 3$.
3. For each $i$-tuple $C$, for each ball in $C$, adjust it's neighbor-ball pointers as to point to the other $i-1$ balls that belong in $C, i = 2, 3$.
4. For each pair of indices $0 < i < j < h$, select $x_{i,j}n$, bins of non complementary underlying literals. Fill each such bin with exactly $i$ balls and set it's exposed load-bin register equal to $(i, j)$. For each such bin of load $i$, fill the bin of it's negated literal with $j$ balls and set it's exposed load-bin register to $(j, i)$. For each such pair of bins with negated pair of underlying literals, mutually adjust their unexposed negation-bin registers.

5. For each index $0 < i < h$, select $x_{i,i}n/2$ bins of non complementary underlying literals. Fill each such bin with exactly $i$ balls and set it's exposed load-bin register equal to $(i, i)$. For each such bin of load $i$, fill the bin of it's negated literal with $i$ balls and set it's exposed load-bin register to $(i, i)$. For each such pair of bins with negated pair of underlying literals, mutually adjust their unexposed negation-bin registers.
6. For each index $0 < i < h$ select $x_{i,h}n$ bins of non complementary underlying literals. Fill each such bin with exactly $i$ balls and set it's exposed load-bin register equal to $(i, h)$. For each such bin of load $i$, fill the bin of it's negated literal with at least $h$ balls and set it's exposed load-bin register to $(h, i)$. For each such pair of bins with negated pair of underlying literals, mutually adjust their unexposed negation-bin registers.
7. Throw the remaining heavy balls into the remaining $x_{h,h}n$ heavy bins, such that, each heavy bin receives at least $h$ balls. For each heavy bin set it's exposed load-bin register to $(h, h)$. For each pair of heavy bins with negated pair of underlying literals, mutually adjust their unexposed negation-bin registers.
8. For each ball$_i, i = 1, \ldots, pn$, landed into bin$_j, j = 1, \ldots, \ell n$, adjust it's negation-ball$_i$ as to point to the bin corresponding to the negation of the literal of bin$_j$.

*Remark 2.* Although the above process may give rise to improper formulas, formulas containing $i$-clauses with complementary literals or repetitions of a literal, observe that it gives rise with positive probability to proper formulas with none of these improper features. Furthermore, all proper formulas are equally likely to appear.

For the probabilistic analysis of algorithm `CL` we need the following Theorem which we prove applying the technique in [24].

**Theorem 2.** *At the end of each round of the algorithm* `CL` *the reduced formula is random given the current* $(h + 1)^2 + 3$ *scaled parameters of the vector* $\mathcal{S}$ *in (1).*

*Proof.* We can express each free/forced step of `CL` as operations on the process (2) using the processes (3) and (4). It is crucial for the randomness invariance that as soon as processes (3), (4) end we obtain the initial type of exposed and unexposed registers.

`Free Step` (3)

1. Specify a pair of indices $(i, j) \in \{0, \ldots, h\}^2$.
2. Point to a bin amongst the ones with *exposed* load-bin register equal to $(i, j)$, say bin $p$.
3. Delete $p$ and it's content of balls.
4. Delete the balls pointed by the neighbor-registers of the balls in $p$.
5. Delete bin $\overline{p}$, pointed by the negation-bin register of $p$, and it's content of balls.

6. Update all the (un)-exposed registers based upon the undeleted balls and bins.

As soon as the free step is complete it is possible some exposed clause-ball registers to become 1. Then a sequence of forced atomic steps starts, such that all 1-clauses are satisfied. We express each forced step in terms of the balls-in-bins model:

`Forced Step` (4)

1. Point to a random ball $b$ amongst the ones with *exposed* clause-ball register equal to 1.
2. Delete the bin $p_b$ of ball $b$ and it's content of balls.
3. Delete the balls pointed by the neighbor-registers of the balls in $p_b$.
4. Delete bin $\overline{p}_b$, pointed by the negation-bin register of $p_b$, and it's content of balls.
5. Update all the (un)-exposed registers based upon the undeleted balls and bins.

Observe that the content of the each unexposed register can not be inferred at the end of each atomic step. We only know the content of each exposed register per atomic step. The result follows from [24].

*Remark 3.* Steps 1-2 of random process (2) and steps 4, 3 in random processes (3) and (4) respectively guarantee that each 1-clause generated within an arbitrary round corresponds to a *completely random* literal occurrence. This is crucial in proving that the generation of 1-clauses corresponds to a Galton Watson branching process, see Section 7.

## 6 Statistics of the Bins

Algorithm `CL` is initialized with a formula having the *typical* double degree sequence:

**Proposition 1.** *A random 3-SAT formula of density c w.h.p. has the typical double degree sequence:*

$$x_{i,j} = \ell e^{-2(3c/\ell)} \frac{(3c/\ell)^{i+j}}{i!j!}, (i,j) \in \{0,\dots,h-1\}^2,\ i,j < h,$$

$$x_{i,h} = x_{h,i} = \ell e^{-(3c/\ell)} \frac{(3c/\ell)^{i}}{i!} \left(1 - \sum_{s=0}^{h-1} e^{-(3c/\ell)} \frac{(3c/\ell)^{s}}{s!}\right),\ i < h,$$

$$x_{h,h} = \ell - \sum_{s,t\geq 0}^{h-1} x_{s,t} - 2\sum_{s=0}^{h-1} x_{s,h}.$$

*Proof.* See Appendix 1.

**Theorem 3.** *Any literal $\tau \in L$ and any literal occurrence $b$ in a formula which is random given the vector $\mathcal{S}$ in (1) have the following properties:*

1. $$P_h(\mu; k) = \Pr[deg(\tau) = k|\ \tau \in \mathcal{X}_{h,j}] = \frac{\mu^k}{\left(e^\mu - \sum_{s=0}^{h-1} \frac{\mu^s}{s!}\right) k!}, k \geq h, j \leq h,$$
$$\lambda_h = \frac{\mu\left(e^\mu - \sum_{j=0}^{h-2} \frac{\mu^j}{j!}\right)}{e^\mu - \sum_{j=0}^{h-1} \frac{\mu^j}{j!}},\ \textit{i.e, we solve with respect to } \mu \textit{ and,}$$
$$\lambda_h = \frac{p - \left(\sum_{s=0,t>s}^{h-1}(s+t)x_{s,t} + \sum_{s=0}^{h-1}(sx_{s,h} + sx_{s,s})\right)}{\sum_{s=0}^{h} x_{s,h}}\ \textit{is the average}$$
*load of a heavy bin.*
2. $\Pr[\exists\ \tau:\ deg(\tau) > \ln n|\ \tau \in \mathcal{X}_{h,j}] \leq e^{-(1-o(1))\ln n \ln\ln n},\ j \leq h.$
3. $\Pr[\tau \in \mathcal{X}_{i,j}] = \frac{x_{i,j}}{\ell}, (i,j) \in A^2.$
4. $$\Pr[\textit{Literal occurrence } b \in \mathcal{X}_{i,j}] = \frac{\zeta_i^h x_{i,j}}{p}, (i,j) \in A^2,$$
$$\textit{we define: } \zeta_i^h = \begin{cases} i, & i < h, \\ \lambda_h, & i = h. \end{cases}$$
5. $m = \mathbf{E}[deg(\overline{b})|\ b \textit{ is a literal occurrence}] = \frac{1}{p}\sum_{s,t=0}^{h} \zeta_s^h \zeta_t^h x_{s,t}.$
6. $$\varepsilon_1 = \mathbf{E}[deg(b) \textit{ in 2, 3-clauses}|\ b \textit{ is a literal occurrence in a 1-clause}]$$
$$= \frac{1}{p}\left(\sum_{s=2}^{h-1}\sum_{t=0}^{h}(s-1)tx_{s,t} + \sum_{s=0}^{h} x_{h,s} \frac{\mu^2 e^\mu - \sum_{s=2}^{h-1} \frac{s(s-1)\mu^s}{s!}}{e^\mu - \sum_{s=0}^{h-1}\frac{\mu^s}{s!}}\right)$$

*Proof.* See Appendix 2.

*Remark 4.* Parts 1, 2 of Theorem 3 guarantee the sharp tails bounds of a Poisson distribution for the heavy literals. This is important in Section 7 in proving that a length of a round is $O(poly \ln^2 n)$. Then we use this in Section 9, in accordance with the sharp tail bound on the load of a bin established here, to satisfy condition $(ii)$ and $(i')$ of Theorems 1 and 2, respectively, in [33].

## 7 Size of a Round & Probability of Success

Consider an arbitrary round and suppose the algorithm during it's free step Selects & Sets to 1 a literal $\tau \in \mathcal{X}_{i,j},\ i,j \in A^2$. If $j < h$ then exactly $j$ balls containing $\overline{\tau}$ are deleted. If $j = h$ then exactly $k \geq h$ balls are deleted with probability $P_h(\mu; k)$, that is $\lambda_h$ in expectation, see part 1 of Theorem 3. Expectedly, $deg(\overline{\tau}) = \zeta_j^h$ literal occurrences (balls) that contain $\overline{\tau}$ are deleted, see process (3) Step 5. According to part 2 of Theorem 3 the probability that $deg(\overline{\tau}) > \ln n$ is exponentially small. From process (2) Step 1, each of these balls belongs into a 2-clause with probability $2c_2/p$. Then $\zeta_j^h \frac{2c_2}{p}$ new 1-clauses (literal occurrences) are expected to generate in the single free step. It is crucial that each of these new 1-clauses corresponds to a random literal occurrence, see Remark 3. From part 5 of Theorem (3), the negation of a literal occurrence appears in $m$ clauses, in expectation. Therefore in each forced step we expect $m_2 = m\frac{2c_2}{p}$ new 1-clauses to be generated. That is, the generation of new 1-clauses is a Galton-Watson branching process (see [10]) and $m_2$ is it's Malthus parameter. If $m_2 < 1$ the branching process becomes extinct eventually, and the total expected number of 1-clauses in this round equals $\zeta_j^h \frac{2c_2}{p} \frac{1}{1-m_2} = O(1)$, independently of the distribution of the offspring. By the sharp tail bounds of the truncated Poisson distribution (part 2 of Theorem (3)) that each heavy literal obeys, it follows that each individual's offspring is $> O(\ln^2 n)$ with exponentially small probability, see Remark 4. Thus, the size of the G-W tree is at most $O(\ln^2 n)$ w.h.p., see Appendix 3.

During a single or, even during $t(n) = o(n)$ rounds, each of the parameters in $\mathcal{S}$ changes by $(1 + o(1))$. This follows from part 2 of Theorem (3), i.e., there is an exponentially small probability to delete more than $\ln n$ balls per atomic step, only effecting each parameter (of order $\Omega(n)$) within o(1) per atomic step, w.h.p. Also, since $m_2 < 1$ the total number of atomic steps is 1+ number of forced steps $< O(\ln^2 n)$ with certainty, per round.

*Remark 5.* The fact that the length of a round is $< O(\ln^2 n)$ and in each step we delete at most $< O(\ln n)$ balls, is important in proving that condition $(i')$ of Theorem 2 in [33] holds for each unscaled parameter corresponding to vector $\mathcal{S}$ (1). It is also important in proving that condition $(ii)$ of Theorem 1 in [33] holds for each expected change in parts $(a) - (d)$ of lemma 2.

Condition on the size $|T|$ of the tree of G-W process during a single round. Let $b$ an occurrence amongst $|T|$ literal occurrences that correspond to 1-clauses. The probability that $\overline{b}$ appears in at least one of these $|T|$ occurrences is $O(\frac{deg(\overline{b})T}{pn}) = O(|T|/n)$ (we want at least one of these $|T|$ balls to appear into the bin of $\overline{b}$, see Steps 4-7 of process (2)). Thus, given $|T|$, the probability of obtaining a 0-clause during this round is $O(|T|^2/n)$. The unconditional probability of contradiction is $O(\frac{1}{n}\sum_{k\geq 0}|k|^2 \Pr[|T| = k]) = O(size_2/n)$, where $size_2$ is the second moment of the size of the G-W tree. Then the probability that no contradiction occurs in all $O(n)$ rounds is $(1 - O(size_2/n))^{O(n)} = e^{-O(size_2)}$. The following lemma implies the positive probability of success.

**Lemma 1.** *During a round such that $m_2 < 1$, the second moment $size_2$ of the size of the G-W tree corresponding to the generation of 1-clauses is finite.*

*Proof.* See Appendix 4.

## 8 Differential Equations

**Lemma 2.** *Suppose that for $O(n)$ rounds the algorithm* `CL` *selects and set to* TRUE *pairs of complementary literals $(\tau, \overline{\tau}) \in \mathcal{X}_{i,j}$, with arbitrary $(i, j) \in A^2$. Then the $(h+1)^2 + 3$ parameters in the vector:*

$$\mathcal{S} = \langle \ell, c_3, c_2, x_0, \ldots, x_{h-1} \rangle,$$

*are given by the solution of the following system of differential equations:*

$$(a)\ \ \frac{d\ell}{dt} = -2\left(1+\varepsilon_{\overline{\tau}}\right),$$

$$(b)\ \ \frac{dc_3}{dt} = -\left(deg(\tau)+deg(\overline{\tau})\right)\frac{3c_3}{p} - \left(\varepsilon_1+m\right)\frac{3c_3}{p}\varepsilon_{\overline{\tau}},$$

$$(c)\ \ \frac{dc_2}{dt} = deg(\overline{\tau})\frac{3c_3}{p} - \left(deg(\tau)+deg(\overline{\tau})\right)\frac{2c_2}{p} + \left(m_3 - \left(\varepsilon_1+m\right)\frac{2c_2}{p}\right)\varepsilon_{\overline{\tau}},$$

$$(d)\ \ \frac{dx_{i,j}}{dt} = deg(\tau)\tfrac{6c_3+2c_2}{p^2}f(i,j) - \delta_{i,j}^{\tau,\overline{\tau}} + \left(\varepsilon_1\tfrac{6c_3+2c_2}{p^2}f(i,j) - g(i,j)\tfrac{x_{i,j}}{p}\right)\varepsilon_{\overline{\tau}},$$
$$\forall\ (i,j)\in\{0,\ldots,h\}^2\setminus(h,h).$$

*where:*

$$\varepsilon_{\overline{\tau}} = deg(\overline{\tau})\frac{2c_2}{p(1-m_2)},\ \ \textit{and } p = 3c_3+2c_2.$$

$$m_k = m\frac{kc_k}{p},\ \ k=2,3,\ \ \textit{also:}$$

$$\delta_{i,j}^{\tau,\overline{\tau}} = \begin{cases} 1, & i\neq j \textit{ and } i=deg(\tau), j=deg(\overline{\tau}) \textit{ or } i=deg(\overline{\tau}), j=deg(\tau),\\ 2, & i=j=deg(\tau)=deg(\overline{\tau}),\\ 0, & \textit{otherwise.}\end{cases}$$

$$f(i,j) = \begin{cases} (i+1)x_{i+1,j}\theta_i^h + (j+1)x_{i,j+1}\theta_j^h - (i+j)x_{i,j}, & i,j<h,\\ (k+1)x_{k+1,h} - kx_{k,h} - hx_{k,h}\theta_{h-1}^h, & (i,j)\in B,\\ hx_{h,h}\theta_{h-1}^h - (h-1)x_{h-1,h} - hx_{h-1,h}\theta_{h-1}^h, & (i,j)\in G,\end{cases}$$

*where:* $B = \{(h,k),(k,h)\}, k\leq h-2,$

*and:* $G = \{(h-1,h),(h,h-1)\},$

*also:* $\theta_s^h = \begin{cases} P_h(\mu;h), & s=h-1,\\ 1, & \textit{otherwise.}\end{cases}$

$$g(i,j) = \begin{cases} i+j, & 0\leq i,j\leq h-1,\\ k+\lambda_h, & (i,j)\in B,\\ h-1+\lambda_h, & (i,j)\in G.\end{cases}$$

*Initial conditions:*

$\ell=2,\ c_3=c,\ c_2=0.00005,$ *and each* $x_{i,j},\ (i,j)\in A^2$ *is as in Proposition 1.*

*Proof.* See Appendix 5.

## 9 Wormald's Theorem

Assume that $Y_j = \Omega(n), j=1,\ldots,a,$ are stochastic parameters related to a formula, like e.g. $|L|, |C_i|, i=2,3,$ etc. Suppose that, given the current vector

$\langle Y_1(t), \ldots, Y_a(t) \rangle$, the expected change of each $Y_j, j = 1, \ldots, a$, from round $t$ to $t+1$ is given by a function $f_j$:

$$\mathbf{E}[Y_j(t+1) - Y_j(t) \mid Y_1(t), \ldots, Y_a(t)] = f_j(t/n, Y_1(t)/n, \ldots, Y_a(t)/n) + o(1),$$

such that $f_j$ has the properties:

1. Each conditional change of random variable $Y_j$ is concentrated to $f_j$.
2. $f_j = O(1)$ in each round, and if we only know each random variable $Y_j(t)$ within $o(n)$ we can determine $f_j$ within o(1). That is, fluctuations within $o(n)$ of each random variable cannot effect it's expected change.
3. Each function $f_j$ is Lipschitz continuous.

Then the solution of the system of differential equations:

$$dy_j(x)/dx = f_j(x, y_1(x), \ldots, y_a(x)), j = 1, \ldots, a,$$

gives almost surely (as $n \to \infty$) in each round $t$ the scaled value of each $Y_j$:

$$y_j(t/n) = (1/n)Y_j(t) + o(1).$$

1. Notice that Remark 5 implies condition $(i')$ of Theorem 2 in [33].
2. For a subcritical degree sequence $\mathcal{S}$ with $m_2 < 1$, the functions $(a) - (d)$ of lemma 2 describe within $o(1)$ the conditional expected changes. In this way we satisfy condition $(ii)$ of Theorem 1 in [33].
3. Since $m_2 < 1$ we obtain that $\varepsilon_{\overline{\tau}}$ is bounded. Furthermore, for $O(n)$ rounds we select and set pairs of literals from a specific bucket. These imply that functions $(a) - (d)$ are Lipschitz continuous as required from condition $(iii)$ Theorem 1 in [33].
4. In each of the $O(|T|)$ steps of the round it is possible to encounter a 0-clause or both $\tau, \overline{\tau}$ to appear in a clause or even both $\tau, \tau$ to appear in the same clause. However, for any given literal $\tau$ the probability that $\overline{\tau}$ occurs in these $O(|T|)$ 1-clauses is $O(|T|/n)$ . Therefore given that the size of the tree of a round is $|T|$ the conditional probability of contradiction is at most $O(|T||T|/n) = O(|T|^2/n)$. The same holds for the other bad events. It follows that the unconditional probability is of order $O(size_2/n)$, where $size_2$ is the second moment of the size of the G-W tree. In lemma 1 it is proved that $size_2 = O(1)$, i.e. such bad events are insignificant for the conditional expected change of each parameter in $\mathcal{S}$.

## 10 Implementation & Termination of the Algorithm

We plugged into the system of lemma 2 initial conditions corresponding to $c = 3.52+$. At round 0 we computed $m_2(0)$ and $m_3(0)$. Recall from part 5 of Theorem 3 and lemma 2, that $m_i(t)$ equals the expected number of unsatisfied $i$-clauses, $i = 2, 3$ in each *forced* step of round $t$. For each $(i, j) \in A^2$, we performed $\epsilon = 1/100000$ rounds (restarting from round 0 each time) and we computed the corresponding $R_1(i, j) = \frac{m_2(\epsilon) - m_2(0)}{m_3(\epsilon) - m_3(0)}$. Let $R_1(i_1, j_1)$ be the maximum. This preprocessing step of computing $R_1(i_1, j_1)$ amounts to the procedure `Choose-Bucket` of algorithm `CL`. Then we restarted from round 0 and for $T_1 = O(n)$ rounds we always set literals from $\mathcal{X}_{i_1,j_1}$. Similarly, we performed $t_1$ rounds (this time starting from round $T_1$ each time) and we computed the corresponding $R_2(i,j) = \frac{m_2(T_1+\epsilon)-m_2(T_1)}{m_3(T_1+\epsilon)-m_3(T_1)}$, $\forall (i,j) \in A^2$. Let $R_2(i_2, j_2)$ be the new maximum (see procedure `Choose-Bucket`). Now, we restarted from round $T_1$ and for $T_2 = O(n)$ rounds we always set literals from $\mathcal{X}_{i_2,j_2}$, etc. We take care that each scaled parameter in $\mathcal{S}$ remains $> 0$. For initial density $c = 3.52$ the Malthus parameter $m_2$ remained $< 1$ always. We simulated the differential equations until we reached a round $t^*$ such that we could apply lemma 3 and safely terminate the algorithm. Using Theorem 4 (see Appendix 7) proved by Cooper Frieze and Sorkin in [9], we prove this lemma.

**Lemma 3.** *A random formula given S in (1) is almost sure satisfiable if it holds:* $2D_2 < (1-\epsilon)D_1'$, *where* $D_1' = 2(c_2 + c_3)n$ *and:*

$$D_2 = \frac{1}{2}\left(\sum_{s,t\geq 0}^{h-1} st x_{s,t} + 2\lambda_h \sum_{s=0}^{h-1} s x_{h,s} + \lambda_h^2 x_{h,h}\right) n.$$

*Recall that* $\lambda_h$*, defined in part 2 of Theorem 3, is the expected load of a heavy bin.*

*Proof.* See Appendix 6.

## References


1. Achlioptas, D.: Setting two variables at a time yields a new lower bound for random 3-SAT. In: Proc. 32nd Annual ACM Symposium on Theory of Computing (STOC '00) 28–37
2. Achlioptas, D.: Lower bounds for random 3-SAT via differential equations. Theoretical Computer Science **265** (1-2) (2001) 159–185
3. Achlioptas, D., Moore, C.: The asymptotic order of the random $k$-SAT threshold. In: Proc. 43rd Annual Symposium on Foundations of Computer Science (FOCS '02)
4. Achlioptas D., Sorkin G.B.: Optimal myopic algorithms for random 3-SAT. In: Proc. 41st Annual Symposium on Foundations of Computer Science (FOCS '00) 590–600
5. Broder, A., Frieze, A., Upfal, E.: On the satisfiability and maximum satisfiability of random 3-CNF formulas. In: Proc. 4th ACM-SIAM Symposium on Discrete Algorithms (SODA '93) 322–330
6. Chao, M.T., Franco, J.: Probabilistic analysis of two heuristics for the 3-satisfiability problem. SIAM J. of Comp. **15** (4) (1986) 1106–1118
7. Chao, M.T., Franco, J.: Probabilistic analysis of a generalization of the unit-clause literal selection heuristics for the $k$-satisfiability problem. Inform. Sci., **51** (3) (1990) 289–314
8. Chvátal, V., Reed, B.: Mick gets some (the odds are on his side). In: Proc. 33rd Annual Symposium on the Foundation of Computer Science (FOCS '92) 620–627

9. Cooper, C., Frieze, A., Sorkin, G.B.: A note on random 2-SAT with prescribed literal degrees. In: Proc. 13th ACM-SIAM Symposium on Discrete Algorithms (SODA '02) 316–320
10. Devroye, L.: Branching processes and their applications in the analysis of tree structures and tree algorithms. In: Habib, M., McDiarmid, C., Alfonsin, R., Reed, B. (eds.): Probabilistic Methods for Algorithmic Discrete Mathematics. Lecture Notes in Computer Science, Vol. . Springer-Verlag, Berlin (1998) 249–314
11. Dubois, O., Boufkhad, Y.: A general upper bound for the satisfiability threshold of random $r$-SAT. J. of Algorithms **24** (1997) 395–420
12. Dubois, O., Boufkhad, Y., Mandler, J.: Typical random 3-SAT formulae and the satisfiability threshold. In: Proc. 11th Symposium on Discrete Algorithms (SODA '00) 126–127
13. Dubois, O., Dequen, G.: A backbone Search Heuristic for Efficient Solving of Hard 3-SAT Formulae. In Proc.: 17th International Joint Conference on Artificial Intelligence (IJCAI '01) 248-253
14. Dubois, O., Dequen, G.: Renormalization as a function of clause lengths for solving $k$-SAT. In: Proc. 5th International Symposium on the Theory and Applications of Satisfiability Testing (SAT 02) 130–132
15. Fernadez, W. de la Vega.: On random 2-SAT. Manuscript, 1992.
16. Friedgut, E., (Appendix by Bourgain, J.): Sharp thresholds of graph properties, and the $k$-SAT problem. J. AMS **12** (1997) 1017–1054
17. Frieze, A., Suen, S.: Analysis of two simple heuristics for random instances of k-SAT. J. Algorithms. **20** (1996) 312–355
18. Frieze A., Wormald, N.: $k$-SAT: a tight threshold for moderately growing $k$. In: Proc. 5th International Symposium on the Theory and Applications of Satisfiability Testing (SAT 02) 1–6
19. Goerdt, A.: A threshold for unsatisfiability. J. Comput. System Sci. **53** (3) (1996) 469-486
20. M.T. Hajiaghayi and G. Sorkin, personal communication, 2002.
21. Janson, S., Stamatiou, Y.C., Vamvakari, M.: Bounding the Unsatisfiability Threshold of Random 3-SAT. Random Structures and Algorithms **17** (2000) 103–116
22. Kamath, A., Motwani, R., Palem, K., Spirakis, P.: Tail bounds for occupancy and the satisfiability threshold conjecture. Random Structures and Algorithms **7** 1995 59–80
23. Kaporis, A.C., Kirousis, L.M., Lalas, E.G.: The probabilistic analysis of a greedy satisfiability algorithm. In: Proc. 10th Annual European Symposium on Algorithms (ESA 2002) Track A 574–585
24. Kaporis, A.C., Kirousis, L.M., Stamatiou, Y.C.: How to prove conditional randomness using the Principle of Deferred Decisions. TR, CTI, Greece, (2002) Available at: www.ceid.upatras.gr/faculty/kirousis/kks-pdd02.ps
25. Kaporis, A.C., Kirousis, L.M., Stamatiou, Y.C., Vamvakari, M., Zito, M.: The Unsatisfiability Threshold Revisited. To appear in Discrete Mathematics
26. Kirousis, L.M., Kranakis, E., Krizanc, D., Stamatiou, Y.C.: Approximating the Unsatisfiability Threshold of Random Formulas. Random Structures and Algorithms **12** (1998) 253–269
27. Maftouhi, M., Fernandez, V.: On random 3-SAT. Combinatorics, Probability and Computing **4** (1995) 190–195
28. Mézard, Parisi, G., M., Zecchina, R.: Analytic and Algorithmic Solution of Random Satisfiability Problems. Science **297** (812) (2002)
29. Mézard, M., Zecchina, R.: The random $k$-Satisfiability problem: from an analytic solution to an efficient algorithm. Phys.Rev. E in press (2002)

30. Mitchel, D., Selman, B., Levesque, H.: Hard and easy distribution of SAT problems. In: Proc. 10th National Conference of Artificial Intelligence (AAAI 1992) 459–465
31. Mitzenmacher, M.: Tight thresholds for the pure literal rule. TR, Digital Equipment Corporation, (1997) Available at: www.research.compaq.com/SRC/
32. Molloy, M.: The probabilistic method. In: Habib, M., McDiarmid, C., Alfonsin, R., Reed, B. (eds.): Probabilistic Methods for Algorithmic Discrete Mathematics. Lecture Notes in Computer Science, Vol. . Springer-Verlag, Berlin (1998) 1–35
33. Wormald, N.C. Differential equations for random processes and random graphs. The Annals of Applied Probability. **5** (4) (1995) 1217–1235

# Appendix 1

## 1.1 Proof of Proposition 1

For a uniformly random formula with $n$ variables and density $c$, each one of the $3cn$ clause positions has probability to be occupied by any literal $\tau \in L$ equal to $1/(\ell n)$. As $n \to \infty$, the probability of literal $\tau \in L$ having degree $i$ follows a Poisson distribution: $P(3c/\ell; i)$. By independence, the same holds for $\overline{\tau} \in L$ having degree $j$ : $P(3c/\ell; j)$. So the expected value of the scaled number of complementary literals of degree $i$ and $j$ respectively, equals: $\ell P(3c/\ell; i)P(3c/\ell; j) = \ell e^{-2(3c/\ell)} \frac{(3c/\ell)^{i+j}}{i!j!}, i, j < h$. Also, the corresponding scaled expectation of complementary literals of degree $i$ and at least $h$ respectively, equals: $\ell P(3c/\ell; i)\left(1 - \sum_{s=0}^{h-1} P(3c/\ell; s)\right) = e^{-(3c/\ell)} \frac{(3c/\ell)^i}{i!} \left(1 - \sum_{s=0}^{h-1} e^{-(3c/\ell)} \frac{(3c/\ell)^s}{s!}\right)$, $i < h$. From a simplified version of Azuma's inequality (see the "Simple Concentration Bound" in [32]) it follows that almost all uniformly random formulas with $n$ variables and density $c$ have truncated scaled double degree sequence with parameters as above.

# Appendix 2

## 2.1 Proof of Theorem 3

*1.* At the end of each round, the load-bin register of each heavy bin only exposes that the load is at least $h$, without revealing it's precise load. This is described in Steps 6, 7 of the `Balls-in-Bins` process in the proof of Theorem 2. Additionally, even in Step 4 and Step 3 of the `Free Step` (3) and `Forced Step` (4) process respectively, where we may reduce the content of a heavy bin by deleting some balls, we reveal it's precise content *only* if it's $< h$. This is equivalent to throw randomly $p_h n = \left(p - \left(\sum_{s=0,t>s}^{h-1} (s+t)x_{s,t} + \sum_{s=0}^{h-1} (sx_{s,h} + sx_{s,s})\right)\right) n$ heavy balls in $x_h n = \sum_{s=0}^{h} x_{h,s} n$ heavy bins such that each bin gets load at least $h$. In [23], it was proved that the probability that a heavy bin has load $k \geq h$ (a heavy literal has degree $k$) obeys the following truncated Poisson distribution (a generalization of the case $h = 1$ proved in [5]):

$$P_h(\mu; k) = \frac{\mu^k}{\left(e^\mu - \sum_{s=0}^{h-1} \frac{\mu^s}{s!}\right) k!}, \ k \geq h, \ \text{where } \mu \text{ is the solution the equation:}$$

$$\lambda_h = \frac{\mu\left(e^{\mu} - \sum_{s=0}^{h-2} \frac{\mu^s}{s!}\right)}{e^{\mu} - \sum_{s=0}^{h-1} \frac{\mu^s}{s!}},\ \lambda_h = p_h/x_h,\ \text{is the expected load of a heavy bin.}$$

*2.* Inequality (4.2) of Theorem 4.2 in [5] applies verbatim in the balls-in-bins model, therefore:

$$\Pr[\exists \text{ Heavy bin with load} > \ln n] \leq e^{-(1-o(1)) \ln n \ln\ln n},$$

i.e., we have sharp concentration to the expected load.

*3.* Literal $\tau \in L$ corresponds to a random bin of $\ell n$ possible and there are exactly $x_{i,j}n$ bins with underlying literals in the set $\mathcal{X}_{i,j},\ (i,j) \in A^2$.

*4.* According to `Balls-in-Bins` random process (2), each literal occurrence $b$ corresponds to a ball, from $pn = (3c_3 + 2c_2)n$ possible, that u.a.r. lands to a bin. If $i < h$, the bins with underlying literals in $\mathcal{X}_{i,j}$ receive randomly $ix_{i,j}n, j \in \{0, \ldots, h\}$, balls, amongst $pn$ possible balls. If $i = h$, consider heavy literal $\tau \in \mathcal{X}_{h,j}$. From part 1 above, $deg(\tau) = k \geq h$ with probability $P_h(\mu; k)$. Therefore there are $x_{h,j}P_h(\mu; k)n$ literals (bins) in $\mathcal{X}_{h,j}$ each having degree $k \geq h$. Their corresponding bins contain exactly $x_{h,j}P_h(\mu; k)kn$ balls. Then in the bins of $\mathcal{X}_{h,j},\ j \in \{0, \ldots, h\}$, there are exactly $\left(\sum_{k=h}^{\infty} x_{h,j}P_h(\mu; k)k\right) n = \lambda_h x_{i,j} n$ balls, amongst $pn$ possible balls.

*5.* If $k < h$ then $\Pr[deg(\overline{b}) = k] = \Pr[b \in \mathcal{X}_{1,k} \cup \ldots \cup \mathcal{X}_{h,k}]$. By part 4 above, $\Pr[deg(\overline{b}) = k] = \frac{1x_{1,k} + 2x_{2,k} + \ldots + \lambda_h x_{h,k}}{p},\ k < h$. If $k \geq h$ then:

$$\begin{aligned}\Pr[deg(\overline{b}) = k] &= \Pr[deg(\overline{b}) = k \wedge (b \in \mathcal{X}_{1,h} \cup \ldots \cup \mathcal{X}_{h,h})] \\ &= \Pr[deg(\overline{b}) = k \ \wedge\ b \in \mathcal{X}_{1,h}] + \ldots + \Pr[deg(\overline{b}) = k \ \wedge\ b \in \mathcal{X}_{h,h}],\end{aligned} \tag{5}$$

since the sets $\mathcal{X}_{j,h}, j = 1, \ldots, h$ are disjoint. From part 4 we obtain:

$$\Pr[deg(\overline{b}) = k \ \wedge\ b \in \mathcal{X}_{j,h}] = \frac{\zeta_j^h x_{j,h}}{p} P_h(\mu; k) \tag{6}$$

Summing (6) over $j = 1, \ldots, h$ and plugging in (5) we obtain for all $k$:

$$\Pr[deg(\overline{b}) = k] = \begin{cases} \frac{1x_{1,k} + 2x_{2,k} + \ldots + \lambda_h x_{h,k}}{p}, & k < h, \\ \frac{1x_{1,h} + 2x_{2,h} + \ldots + \lambda_h x_{h,h}}{p} P_h(\mu; k), & k \geq h. \end{cases} \tag{7}$$

Using (7) we obtain:

$$\begin{aligned}\mathbf{E}[deg(\overline{b})] &= \sum_{k=0}^{h-1} k \Pr[deg(\overline{b}) = k] + \sum_{k=h}^{\infty} k \Pr[deg(\overline{b}) = k] \\ &= \frac{1}{p} \sum_{k=0}^{h-1} \sum_{i=1}^{h} k\zeta_i^h x_{i,k} + \frac{1}{p} \sum_{k=h}^{\infty} kP_h(\mu; k) \sum_{i=1}^{h} \zeta_i^h x_{i,h}\end{aligned}$$

$$= \frac{1}{p}\sum_{k=0}^{h-1}\sum_{i=1}^{h} k\zeta_i^h x_{i,k} + \frac{1}{p}\zeta_h^h \sum_{i=1}^{h}\zeta_i^h x_{i,h} = m.$$

*6.* First consider the case that the 1-clause literal occurrence $b$ appears in the 3, 2-clauses exactly $s-1$ times, with $s < h$ (it's occurrence in the 1-clause is subtracted). This occurs iff $b \in \mathcal{X}_{s,1} \cup \ldots \cup \mathcal{X}_{s,h}$. These events are disjoint. That is, if $s < h$ then $b$ appears in $s-1$ other clauses, excluding it's 1-clause, with probability $\Pr[b \in \mathcal{X}_{s,1}] + \ldots + \Pr[b \in \mathcal{X}_{s,h}] = \frac{1}{p}\sum_{t=0}^{h} s x_{s,t}$, since each literal in $\mathcal{X}_{s,t}, 0 \le t \le h$, corresponds to a bin with exactly $s$ balls. In this case we obtain the term:

$$\frac{1}{p}\sum_{s=2}^{h-1}\sum_{t=0}^{h}(s-1)t x_{s,t}. \tag{8}$$

It remains to compute the expected occurrences of $b$ in 3, 2-clauses excluding it's 1-clause, that is to compute the expected $deg(b)-1$, when $b \in \mathcal{X}_{h,0} \cup \ldots \cup \mathcal{X}_{h,h}$. The above events are disjoint:

$$\begin{aligned} pr(k-1) &= \Pr[deg(b) = k \ \wedge \ (b \in \mathcal{X}_{h,0} \cup \ldots \cup \mathcal{X}_{h,h})] \\ &= \Pr[deg(b) = k \ \wedge \ b \in \mathcal{X}_{h,0}] + \ldots + \Pr[deg(b) = k \ \wedge \ b \in \mathcal{X}_{h,h}]. \end{aligned}$$

Consider the event: $\{deg(b) = k \wedge b \in \mathcal{X}_{h,s}\}, 0 \le s \le h$. There are $x_{h,s}P_h(\mu;k)n$ literals (bins) in $\mathcal{X}_{h,s}$ containing $k \ge h$ balls each. Then $b$ appears in these $x_{h,s}P_h(\mu;k)n$ bins of $\mathcal{X}_{h,s}$ with probability $\frac{1}{p}x_{h,s}P_h(\mu;k)k$ which equals the probability $\Pr[deg(b) = k \ \wedge \ b \in \mathcal{X}_{h,s}], 0 \le s \le h$. Therefore $b$ appears in $k-1$ other 3, 2-clauses with probability: $pr(k-1) = \frac{1}{p}P_h(\mu;k)k\sum_{s=0}^{h} x_{h,s}$. Summing over $k \ge h$ we get:

$$\begin{aligned} \sum_{k=h}^{\infty}(k-1)pr(k-1) &= \frac{1}{p}\sum_{s=0}^{h} x_{h,s}\sum_{k=h}^{\infty}(k-1)kP_h(\mu;k) \\ &= \frac{1}{p}\sum_{s=0}^{h} x_{h,s}\frac{\mu^2 e^{\mu} - \sum_{s=2}^{h-1}\frac{s(s-1)\mu^s}{s!}}{e^{\mu} - \sum_{s=0}^{h-1}\frac{\mu^s}{s!}} \end{aligned} \tag{9}$$

Summing (8), (9) we get $\varepsilon_1$.

# Appendix 3

### 3.1 Proof that the size of each round is polylogarithimic

Consider a literal occurrence $b$ corresponding to a 1-clause generated in this round. Recall Equation (7):

$$\Pr_{uc}[deg(\overline{b}) = k] = \begin{cases} \frac{1x_{1,k}+2x_{2,k}+\ldots+\lambda_h x_{h,k}}{p} = f(k) = O(1), & k < h, \\ \frac{1x_{1,h}+2x_{2,h}+\ldots+\lambda_h x_{h,h}}{p} P_h(\mu;k) \le P_h(\mu;k), & k \ge h. \end{cases}, \tag{10}$$

where $P_h(\mu;k)$ is the truncated Poisson distribution in part 1 of Theorem 3. Since each of the $k$ occurrences of $\overline{b}$ is in a 2-clause with probability $2c_2/p$ the distribution in (10) gives an upper bound to the probability that the deletion of $\overline{b}$ generates $k$ 1-clauses. Then there exists a Poisson distribution with mean $\mu^* \geq m$ (see part 5 of Theorem 3) such that:

$$\Pr_{uc}[deg(\overline{b}) = k] \leq P(\mu^*;k), \forall k \geq h.$$

Since Poisson distribution $P(\mu^*;k)$ dominates stochastically the actual offspring of the 1-clauses, the proof follows from Lemma 11 in [4].

## Appendix 4

### 4.1 Proof of Lemma 1

Let $y(s)$ denote the probability generating function of the size $|T|$ of the tree and $f(s)$ the probability generating function of the number $Z$ of offspring. In [10] Theorem 5.8 it is proved that $y(s) = sf(y(s))$. Differentiating this with respect to $s$ we obtain that $size_2$ can be written in terms of $m_2 = O(1)$ and in terms of the second moment of the number of offspring which is $O(\varepsilon_1) = O(1)$.

## Appendix 5

### 5.1 Proof of Lemma 2

$(a)$ In Section 7 it was proved that the expected number of forced steps equals: $\varepsilon_{\overline{\tau}}$. Therefore, we expect $(1 + \varepsilon_{\overline{\tau}})$ steps in the round, setting exactly 2 literals per step.

$(b)$ In the free step we have a total of $deg(\tau) + deg(\overline{\tau})$ occurrences (balls) of the selected pair of literals $\tau, \overline{\tau}$. From steps 1, 2 of process 2 each such ball appears in a 3-clause (deleting or shrinking it) with probability $3c_3/p$.

In each of the expectedly $\varepsilon_{\overline{\tau}}$ forced steps, we select a literal occurrence $b$ that corresponds to 1-clause. According to part 6 of Theorem 3 $b$ is expected to appear in $\varepsilon_1 \frac{3c_3}{p}$ other balls in 3-clauses. From part 5 of Theorem 3 it's negation $\overline{b}$ is expected to appear in $\varepsilon_1 \frac{3c_3}{p}$ other balls in 3-clauses. So in the forced steps of the round we expect to loose $(\varepsilon_1 + m) \frac{3c_3}{p} \varepsilon_{\overline{\tau}}$ 3-clauses.

$(c)$ The line of thought is the same as in part $(b)$ above. Notice that the expected number of 3-clauses that become 2-clause during the single free step equals: $deg(\overline{\tau})\frac{3c_3}{p}$.

In each forced step we expect $m_3$ 3-clauses to become 2-clauses by the deletion of $\overline{b}$ and, we perform $\varepsilon_{\overline{\tau}}$ forced steps in expectation.

$(d)$ In the free step we expect to delete $deg(\tau)\frac{2 \cdot 3c_3 + 2c_2}{p}$ literal occurrences (balls) sharing the same clause with literal occurrence $\tau$.

In each forced step the 1-clause literal occurrence $b$ is expected to delete $\varepsilon_1 \frac{2\cdot 3c_3+2c_2}{p}$ balls sharing the same clause with $b$. It is easy, but tedious to show that each such deleted neighboring ball causes an expected change into $\mathcal{X}_{i,j}$ that equals $\frac{f(i,j)}{p}$.

In the single free step, the operation of indicator variable $\delta_{i,j}^{\tau,\overline{\tau}}$ is explained in Remark 1.

In each forced step, if $i, j < h$ then each 1-clause literal occurrence $b$ belongs to $\mathcal{X}_{i,j}$ with probability $ix_{i,j}/p$ or belongs to $\mathcal{X}_{j,i}$ with probability $jx_{i,j}/p$. If $(i, j) \in B$ then similarly each $b$ belongs to $\mathcal{X}_{i,j}$ with probability $kx_{i,j}/p$ or belongs to $\mathcal{X}_{j,i}$ with probability $\lambda_h x_{i,j}/p$. If $(i, j) \in G$ then similarly $b$ belongs to $\mathcal{X}_{i,j}$ with probability $(h-1)x_{i,j}/p$ or belongs to $\mathcal{X}_{j,i}$ with probability $\lambda_h x_{i,j}/p$.

# Appendix 6

## 6.1 Proof of Lemma 3

Consider a random formula $\phi$ given the current degree sequence $\mathcal{S}$. From part 2 of Theorem 3 it holds w.h.p. that the maximum degree $\Delta_d$ of any literal in $\mathcal{S}$ is at most $\ln n < n^{\alpha}, \alpha < 1/13$. Since the current formula $\phi$ consists of 3, 2-clauses, delete exactly one random literal occurrence (ball) from each 3-clause. Such deletions are feasible since clause-ball registers are exposed, see Definitions 2, 3. This results in a 2-SAT formula $\phi'$ with $(c_3 + c_2)n$ 2-clauses. Almost sure satisfiability of $\phi'$ implies almost sure satisfiability of $\phi$. $\phi'$ is random given a new degree sequence $\mathcal{S}'$. Since we delete literal occurrences, then for each variable $x_i$ it's *new* numbers of positive $d'_i$ and negative $\overline{d}'_i$ occurrences are at most equal to it's old $d_i, \overline{d}_i$. That is, $D_2 = \sum_{i=1}^{n} d_i\overline{d}_i \geq D'_2 = \sum_{i=1}^{n} d'_i\overline{d'}_i$. The new total number of literal occurrences is $D'_1 = 2(c_3 + c_2)n$. Since $D_2 \geq D'_2$ we obtain the useful inequality:

$$2\frac{D_2}{D'_1} \geq 2\frac{D'_2}{D'_1}, \tag{11}$$

allowing us to compute easily $D_2$ instead of $D'_2$.

To compute $D_2$ we work as follows:
For $t, s \in N$ define: $Var_{t,s} = |\{x_i \in V \mid x_i \in \mathcal{X}^{\sharp}_{t,s}\}|$ where $\mathcal{X}^{\sharp}_{t,s}$ consists of literals with degree exactly $t$ while their complements have degree $s$.

$$\begin{aligned} D_2 = \sum_{i=1}^{n} d_i\overline{d_i} = \sum_{s,t} stVar_{s,t} &= \sum_{s=0}^{h-1}\sum_{t=0}^{h-1} stVar_{s,t} + \sum_{s=h}^{\infty}\sum_{t=0}^{h-1} stVar_{s,t} \\ &+ \sum_{s=0}^{h-1}\sum_{t=h}^{\infty} stVar_{s,t} + \sum_{s=h}^{\infty}\sum_{t=h}^{\infty} stVar_{s,t} \end{aligned} \tag{12}$$

If $s, t < h$ and $x_i \in V$, then

$$\Pr\{x_i \in \mathcal{X}^{\sharp}_{t,s}\} = \frac{x_{s,t}}{\ell} = \frac{Var_{s,t}}{(\ell/2)n} \Leftrightarrow Var_{s,t} = \frac{x_{s,t}}{2}n \tag{13}$$

If $s \geq h, t < h$ (by symmetry of indices the case $s < h, t \geq h$ is similar) then

$$\Pr\{x_i \in \mathcal{X}^{\sharp}_{t,s}\} = \frac{x_{h,t} P_h(\mu; s)}{\ell} = \frac{Var_{s,t}}{(\ell/2)n} \Leftrightarrow Var_{s,t} = \frac{x_{h,t} P_h(\mu; s)}{2} n \tag{14}$$

It remains the case $s \geq h, t \geq h$:

$$\begin{aligned} \Pr\{x_i \in \mathcal{X}^{\sharp}_{t,s}\} &= \frac{x_{h,h} P_h(\mu; s) P_h(\mu; t)}{\ell} = \frac{Var_{s,t}}{(\ell/2)n} \\ &\Leftrightarrow Var_{s,t} = \frac{x_{h,h} P_h(\mu; s) P_h(\mu; t)}{2} n \end{aligned} \tag{15}$$

We plug (13), (14), (15) into the corresponding sums of (12). Notice that the expectation $\sum_{s=h}^{\infty} s P_h(\mu; s)$ of the truncated Poisson distribution equals $\lambda_h$. Then we easily obtain $D_2$ that appears in lemma 3.

# Appendix 7

## 7.1 Theorem 4 in [9]

**Theorem 4.** *Let $\phi$ be a random 2-SAT formula given the degree sequence $d = d_1, \overline{d_1}, \ldots, d_n, \overline{d_n}$ that is, literal $x_i, \neg x_i$ has exactly $d_i, \overline{d_i}, i = 1, \ldots, n$ occurrences in $\phi$. Also let*

$$\begin{aligned} \Delta_d &= max\{d_1, \overline{d_1}, \ldots, d_n, \overline{d_n}\} \leq n^{\alpha}, \alpha < 1/13, \\ D_1 &= \sum_{i=1}^{n} (d_i + \overline{d_i}) = 2m, \\ D_2 &= \sum_{i=1}^{n} d_i \overline{d_i}. \end{aligned}$$

1. *If $2D_2 < (1-\epsilon)D_1$ then $\mathbf{Pr}[\phi$ is satisfiable$] \to 1$.*
2. *If $2D_2 > (1+\epsilon)D_1$ then $\mathbf{Pr}[\phi$ is satisfiable$] \to 0$, where $0 < \epsilon < 1$.*