q-Overlaps in the Random Exact Cover Problem

Gabriel Istrate [a,*] Romeo Negrea [b]

[a] *eAustria Institute, V. Pârvan 4, cam 045B, Timişoara, RO 300223, Romania*

[b] *Department of Mathematics, Universitatea Politehnica din Timişoara, Victoriei 2, 300006, Timişoara, Romania*

Abstract

We prove lower and upper bounds for the thresold of the following problem: given $q \in (0,1)$ and $c > 0$ what is the probability that a random instance of the k-Exact Cover problem [KM05] has two solutions of overlap $qn \pm o(n)$?

This problem is motivated by the connection with the *one-step replica symmetry breaking* approach of Statistical Physics.

Key words: exact cover, overlap, probabilistic method.

1 Introduction

The study of *phase transitions in Combinatorial Optimization problems* [PIM06], [HW05] has recently motivated (and brought to attention) the geometric structure of the solution space of a combinatorial problem. Methods such as the *cavity method* and assumptions such as *replica symmetry* and *one step replica symmetry breaking* make significant predictions on the geometry of solution space that are a source of inspiration (and a challenge) for rigorous work.

A remarkable recent advance in this area is due to Mézard et al. [MMZ05], [DMMZ08]. These papers have provided rigorous evidence that for the random k-satisfiability problem (with sufficiently large k) the intuitions concerning the geometry of the solution space provided by the 1-RSB approach are correct.

* Corresponding author

Email addresses: gabrielistrate@acm.org, negrea@math.uvt.ro (Romeo Negrea).

The evidence is based the support of the overlap distribution, shown to be discontinuous via a study of threshold properties for the q-overlap versions of k-SAT.

The approach in this paper is based on the same idea. We study the overlap distribution of the *random k-Exact Cover* problem. The phase transition in this problem has been studied in [KM05]. Zdeborová et al. [RSZ07],[MMR$^+$07] have applied nonrigorous methods from Statistical Physics (the cavity approach) and have suggested that the *1-step Replica Symmetry Breaking* assumption is valid. This motivates us to study the problem q-overlap k-Exact Cover (defined below), and prove lower and upper bounds on its satisfiability threshold.

2 Preliminaries

Definition 1 *An instance* Φ *of the* k-Exact Cover *is specified by a set of* boolean variables $V = \{x_1, \ldots, x_n\}$ *and a set of subsets of size* k *(called* clauses*) of* V*. Instance* Φ *is satisfiable if there is an assignment* A *of variables in* V *that makes* exactly one *variable in each clause evaluate to TRUE.*

Definition 2 *The* Hamming distance *between two assignments* A *and* B*, on* n *variables is* $d_{A,B} = \frac{n}{2} - \frac{1}{2}\sum_{i=1}^{n} A(x_i)B(x_i)$*. The overlap of truth assignments* A *and* B *is the fraction of variables on which the two assignments coincide, that is*

$$overlap(A, B) = \frac{\{i | A(x_i) = B(x_i)\}}{n}.$$

Definition 3 *Let* $q \in (0, 1)$*. The* q*-overlap* k*-Exact Cover is a decision problem specified as follows:*

INPUT: *an instance* F *of* k*-Exact Cover with* n *variables.*

DECIDE: *whether* F *has two assignments* A *and* B *such that*

$$overlap(A, B) \in [q - \varepsilon(n)n^{-1}, q + \varepsilon(n)n^{-1}]. \quad (1)$$

We will refer to a pair (A, B) *as in equation (1) as* satisfying assignments of overlap approximately q*.*

If A, B are two satisfying assignments and $i, j \in \{0, 1\}$ we will use notation $A = i, B = j$ ($A = B = i$, when $i = j$) as a shorthand for $\{x : A(x) = i, B(x) = j\}$.

Definition 4 *For $k \geq 3$, $q \in (0,1)$ define*

$$q_k = \frac{\sqrt{(k-1)(k-2)}}{2+\sqrt{(k-1)(k-2)}}, \tag{2}$$

and

$$\lambda_{q,k} := \begin{cases} \frac{(k-2)q+2+\sqrt{((k-2)q+2)^2+(k-1)^2(k-2)(1-q)^2}}{2(k-1)} & \text{if } q \in (0,q_k), \\ q & \text{otherwise.} \end{cases} \tag{3}$$

Note that for $q < q_k$ the expression for $\lambda_{q,k}$ is the unique positive root of equation

$$k-2+\frac{x(q-2x)}{(k-1)(\frac{1-q}{2})^2+x(q-x)} = 0, \tag{4}$$

and is strictly less than q. Also, $\lambda_{q,k} > q/2$, since for $x \leq q/2$ the numerator of the fraction in equation (4) is positive.

Definition 5 *For $k \geq 3$, $q \in (0,1)$ define $F_{k,q} : (q/2, \lambda_{q,k}) \to (0,\infty)$ by*

$$F_{k,q}(x) = \frac{\ln(\frac{x}{q-x})}{k-2+\frac{x(q-2x)}{(k-1)(\frac{1-q}{2})^2+x(q-x)}} \tag{5}$$

Note that $F_{k,q}$ is well defined, monotonically increasing, and that $\lim_{x\to q/2} F_{k,q}(x) = 0$, $\lim_{x\to\lambda_{q,k}} F_{k,q}(x) = \infty$. Thus function $F_{k,q}$ is a bijection. Denote by $G_{k,q}(x)$ its inverse.

3 Results

We first remark that a simple application of the main result in [Ist07] shows that the problem q-overlap k-Exact Cover has a sharp threshold. Our main result gives lower and upper bounds on the location of this threshold:

Theorem 1 *Let $k \geq 3$ and let $r_{up}(q,k)$ be the smallest $r_* > 0$ such that $\forall r > r_*$*

$$r\ln(P_k(G_{k,q}(r),(1-q)/2,(1-q)/2,q-G_{k,q}(r))) - G_{k,q}(r)\ln(G_{k,q}(r)) - \\ -(q-G_{k,q}(r))\ln(q-G_{k,q}(r)) - (1-q)\ln((1-q)/2) \leq 0.$$

Also let

$$r_{lb}(q) = \begin{cases} \frac{1}{6}\left[\frac{1}{(1-q)^2}-1\right] & \text{for } q < 1-\frac{1}{\sqrt{2}}, \\ \frac{1}{6} & \text{otherwise.} \end{cases} \tag{6}$$

- *For $r > r_{up}(q,k)$ a random instance of q-overlap k-Exact Cover with has, with probability $1-o(1)$, no satisfying assignments of overlap approximately q.*
- *For $r < r_{lb}(q)$ a random instance of q-overlap 3-Exact Cover with has, with probability $1-o(1)$ two satisfying assignments of overlap approximately q.*

4 The upper bound

Our proofs rely on the following fundamental observation:

Lemma 1 *Let A, B be two satisfying assignments, and let C be a clause of length k. Denote by c_0, c_1, c_2, c_3 the number of variables of C in the sets $A = B = 0$, $A = 0, B = 1$, $A = 1, B = 0$, $A = B = 1$ respectively. Clause C is satisfied by both A and B if and only if*

$$\begin{cases} c_0 = k-2,\ c_1 = c_2 = 1,\ c_3 = 0 \\ \ or \\ c_0 = k-1,\ c_1 = c_2 = 0,\ c_3 = 1 \end{cases} \tag{7}$$

Proof.

The conditions that both A and B satisfy C are written as

$$\begin{cases} c_0 + c_1 = k-1,\ c_2 + c_3 = 1 \\ c_0 + c_2 = k-1,\ c_1 + c_3 = 1, \end{cases} \tag{8}$$

a system whose solutions are those from equation (7). □

An immediate consequence of Lemma 1 is that the probability that a pair of assignments satisfies a random instance of k-EC depends only on numbers c_0, c_1, c_2, c_3:

Lemma 2 *Let c_0, c_1, c_2, c_3 be nonnegative numbers. Then*

$$Pr[A, B \models \Phi \mid |A = B = 0| = c_0, \ldots |A = B = 1| = c_3] = P^*(c_0, c_1, c_2, c_3)^m,$$

where

$$P^*(a,b,c,d) = \frac{\binom{a}{k-2}\binom{b}{1}\binom{c}{1} + \binom{a}{k-1}\binom{d}{1}}{\binom{n}{k}} = \frac{\binom{a}{k-2}}{\binom{n}{k}}\left[bc + \frac{(a-k+2)}{(k-1)}d\right] \tag{9}$$

For the upper bound we employ the first moment method. Let $Z = Z(q,F)$ be a random variable defined as

$$Z(q,F) = \sum_{A,B} \delta[|d_{A,B} - nq| \leq e(n)] \cdot \mathbf{1}_{\mathcal{S}(\mathbf{F})}(\mathbf{A}) \cdot \mathbf{1}_{\mathcal{S}(\mathbf{F})}(\mathbf{B}). \tag{10}$$

where $F = F_k(n, rn)$ is a random formula on n variable over $m = rn$ clauses by the size k, the set $\mathcal{S}(F)$ is the set of the EC-assignments to this formula.

Then:

$$E[Z(q,F)] = \sum_{A,B} \delta[|d_{A,B} - nq| \leq e(n)] \cdot Pr[A,B \models F]. \tag{11}$$

For fixed values a, b, c, d there are

$$\binom{n}{a\ b\ c\ d} = \frac{n!}{a! \cdot b! \cdot c! \cdot d!} \text{ pairs of assignments of type } (a,b,c,d).$$

If we denote $\lambda \stackrel{not}{=} a + d = nq \pm \varepsilon(n)$ and $\mu \stackrel{not}{=} b + c = n - \lambda$ then the system

$$\begin{cases} a + d = \lambda \\ b + c = n - \lambda \end{cases}$$

has at most $(\lambda+1)(n-\lambda+1)$ solutions in the set of nonnegative integers. Therefore, the number of quadruples (a,b,c,d) in the sum $E[Z]$ is at most

$$\sum_{\lambda=nq-\varepsilon(n)}^{nq+\varepsilon(n)} (\lambda+1)(n-\lambda+1) = \frac{1}{3}(1+2\varepsilon(n))(3-\varepsilon(n)-\varepsilon(n)^2+3n+3nq-3q^2) \stackrel{def}{=} M.$$

So

$$P[Z>0] \le E[Z] \le M \cdot \max_{(a,b,c,d)} \binom{n}{a\ b\ c\ d} \cdot P^*(a,b,c,d)^{rn}$$

Denote $\alpha = \frac{a}{n}, \beta = \frac{b}{n}, \gamma = \frac{c}{n}, \delta = \frac{d}{n}$. Applying Stirling's formula $n! = (1 + o(1)) \cdot \left(\frac{n}{e}\right)^n \sqrt{2\pi n}$, and also noting that

$$P^*(a,b,c,d) \le (1 + \frac{O(1)}{n}) \cdot P(\alpha,\beta,\gamma,\delta),$$

with

$$P(\alpha,\beta,\gamma,\delta) = \alpha^{k-2}k(k-1)(\beta\gamma + \frac{\alpha\delta}{k-1})$$

we get

$$P[Z>0] \le M \cdot \theta(1) \cdot \max_{(\alpha,\beta,\gamma,\delta)} \cdot \Big[\Big(\frac{1}{\alpha^\alpha\beta^\beta\gamma^\gamma\delta^\delta}\Big) \cdot P(\alpha,\beta,\gamma,\delta)^r\Big]^n$$

Define

$$g_r(\alpha,\beta,\gamma,\delta) = \frac{P(\alpha,\beta,\gamma,\delta)^r}{\alpha^\alpha\beta^\beta\gamma^\gamma\delta^\delta}$$

Lemma 3 *For any $r > 0$ we have*

$$\max\Big\{g_r(\alpha,\beta,\gamma,\delta) : \alpha+\delta = q, \beta+\gamma = 1-q, \alpha,\beta,\gamma,\delta \ge 0\Big\} = g_r(\alpha_*,\beta_*,\gamma_*,\delta_*),$$

with

$$\begin{cases} \alpha_* = G_{k,q}(r), \\ \beta_* = \gamma_* = (1-q)/2, \\ \delta_* = q - G_{k,q}(r). \end{cases} \tag{12}$$

Proof.

First, it is easy to see that

$$g_r(\alpha,\beta,\gamma,\delta) \le g_r\Big(\alpha,\beta_*,\gamma_*,\delta\Big). \tag{13}$$

Indeed, function $x\ln(x)$ is convex, having the second derivative positive, and e^x is increasing so, by Jensen's inequality,

$$\beta^\beta\gamma^\gamma = e^{\beta\ln(\beta)+\gamma\ln(\gamma)} \ge e^{(\beta+\gamma)\ln(\frac{\beta+\gamma}{2})} = \Big(\frac{\beta+\gamma}{2}\Big)^{\beta+\gamma} = \beta_*^{\beta_*}\gamma_*^{\gamma^*}.$$

On the other hand since $\beta\gamma \le \left(\frac{\beta+\gamma}{2}\right)^2 = \beta_*\gamma_*$, we have $P(\alpha,\beta,\gamma,\delta) \le P\Big(\alpha,\beta_*,\gamma_*,\delta\Big)$ and equation (13) follows.

Also

$$g_r\Big(\alpha, \beta_*, \gamma_*, \delta\Big) \le g_r\Big(\alpha_*, \beta_*, \gamma_*, \delta_*\Big) \tag{14}$$

Indeed, replacing $\delta = q - \alpha$, the expression

$$\begin{aligned} t(\alpha) &= \ln g_r(\alpha, \beta_*, \gamma_*, q-\alpha) = \\ &= r \ln\Big(P(\alpha, \beta_*, \gamma_*, q-\alpha)\Big) - \alpha\ln(\alpha) - (q-\alpha)\ln(q-\alpha) - \beta_*\ln(\beta_*) - \gamma_*\ln(\gamma_*) \end{aligned}$$

is a function of α whose derivative is

$$\begin{aligned} t'(\alpha) &= r\frac{P'(\alpha, \beta_*, \gamma_*, q-\alpha)}{P(\alpha, \beta_*, \gamma_*, q-\alpha)} - \ln(\alpha) - 1 + \ln(q-\alpha) + 1 = \\ &= r\frac{(k-1)(k-2)\Big(\frac{\beta+\gamma}{2}\Big)^2 + (k-1)\alpha(q-\alpha) - \alpha^2}{(k-1)\Big(\frac{\beta+\gamma}{2}\Big)^2 + \alpha(q-\alpha)} + \ln(\frac{q-\alpha}{\alpha}). \end{aligned}$$

Taking into account that $\beta + \gamma = 1 - q$ we get

$$t'(\alpha) = r\frac{(k-1)(k-2)\Big(\frac{1-q}{2}\Big)^2 + (k-1)\alpha(q-\alpha) - \alpha^2}{(k-1)\Big(\frac{1-q}{2}\Big)^2 + \alpha(q-\alpha)} + \ln(\frac{q-\alpha}{\alpha}).$$

so $t(\alpha)$ has a maximum on $[0, q]$ at $\alpha_* = \alpha(q, r, k)$ which is a solution of equation

$$r(k-2) + \frac{r\alpha(q-2\alpha)}{(k-1)\Big(\frac{1-q}{2}\Big)^2 + \alpha(q-\alpha)} + \ln(\frac{q-\alpha}{\alpha}) = 0,$$

or $F_{k,q}(x) = r$. In other words $\alpha_* = G_{k,q}(r)$ and $\delta_* = q - G_{k,q}(r)$. $\square$

Formula (12) implies that $P[Z > 0] \overset{n\to\infty}{\to} 0$ as long as $t(\alpha_r) < 0$. The critical line $r_{up}(q, k)$ is therefore given by equation

$$\begin{aligned} &r\ln(P_k(G_{k,q}(r), (1-q)/2, (1-q)/2, q - G_{k,q}(r))) - G_{k,q}(r)\ln(G_{k,q}(r)) - \\ &- (q - G_{k,q}(r))\ln(q - G_{k,q}(r)) - (1-q)\ln((1-q)/2) = 0. \end{aligned}$$

$\square$

For $k = 3$, symbolic and numeric manipulations of the function in the equation defining $r_{up}(q)$ yield the graph plotted in Figure 4. In particular, the maximum

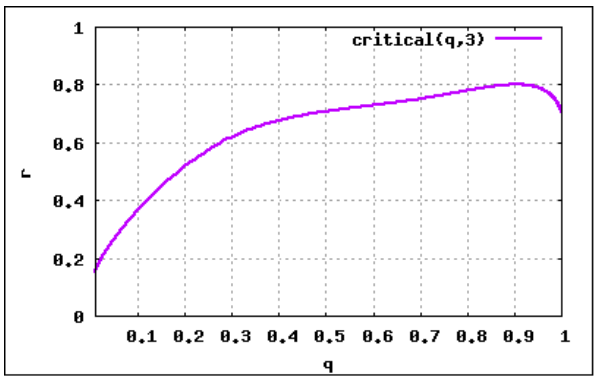

Fig. 1. The upper bound for q-3-Exact Cover.

value of $r_3(q)$ is approximately 0.8, close to upper bound the threshold for 3-Exact Cover derived using the first-moment method in [KSM04].

5 Lower bound

We will use a constructive method. Just as in [KM05], we will derive a lower bound from the probabilistic analysis of an algorithm. However, the algorithm **will not** be the one from [KM05]. Instead, we will investigate the algorithm LARGEST-CLAUSE in Figure 2.

Intuitively, the reason we prefer the algorithm LARGEST-CLAUSE and not the one from [KM05] is simple: we would like an algorithm that iteratively assigns values to variables and is left with a 2-XOR SAT formula. Our aim is to keep the number of set variables to a minimum. But that means that one must "destroy" all clauses of length different from two as fast as possible. Instead, the algorithm in [KM05] is focused on killing clauses of length 2.

To analyze Algorithm LARGEST-CLAUSE, we denote by $C_i(t)$, $i \geq 2$, the number of clauses of length i that are present after t variables have been set. Also define $P(t), N_t$ to be the number of positive (negative) unit clauses present at stage t.

Finally, define functions $c_1, c_2, c_3, p, n : (0,1) \to \mathbf{R}_+$ by

$$c_i(\alpha) = C_i(\alpha \cdot n)/n,$$

and similar relations for functions $p(\cdot), n(\cdot)$. We will use a standard method, *the principle of deferred decisions* to analyze algorithm LARGEST-CLAUSE. See [Ach01] for a tutorial.

It is easy to show by induction that at any stage t, conditional on the four-tuple $(P(t), N(t), C_2(t), C_3(t))$, the remaining formula is uniform.

Algorithm **LargestClause**

INPUT: a formula Φ

```
if (Φ contains a unit clause)
    choose a random unit clause l
    set l to TRUE
        if this creates a contradiction FAIL
        else call the algorithm recursively
else if (Φ contains a clause of length ≥ 3)
        choose a random clause C of maximal length
        choose a random literal l of C
        set l to zero and simplify the formula
    else
            create a graph G containing an edge (x, y)
            for any clause x ⊕ y in Φ;
            if (G is not bipartite) OR (some connected component has size ≥ f(n))
                FAIL
            else
                choose one variable in each connected component of G
                create satisfying assignments A and B
                by setting all chosen variables to one (zero)
                and then propagating these values to all variables in G.
                return (A, B).
```

Fig. 2. Algorithm LARGEST-CLAUSE

We divide the algorithm in two phases: in the *first phase* there exist clauses of length three. In the *second phase* only clauses of length one and two exist.

If a variable is set to TRUE then a 1-in-i clause containing that variable is turned into $i-1$ negative unit clauses. If a variable is set to FALSE then a 1-in-i clause is turned into a 1-in-$(i-1)$ clause, in particular a 1-in-2 clause is turned into a positive unit clasue. The dynamics is displayed in Figure 5.

The different dynamics of the flows in the cases when a positive (negative) literal is set makes the direct analysis of algorithm LARGEST-CLAUSE difficult. Instead we will use a version of the algorithm using a "lazy-server" [Ach01] idea. Specifically, instead of always trying to simplify the unit clauses, we will do so probabilistically. The modified version of algorithm LARGEST-CLAUSE is given in Figure 4.

Let $U_P(t), U_N(t), U_3(t)$ be 0/1 variables that are one exactly when choice 1 (2,3) is selected. We can write the following recurrence relations describing

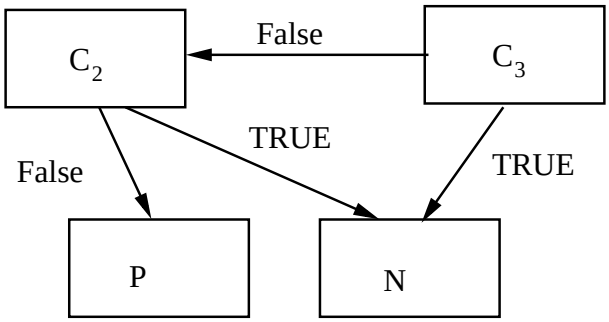

Fig. 3. The dynamics of algorithm LARGEST-CLAUSE.

Algorithm **LazyLargestClause**

INPUT: a formula Φ

if (at least one of the alternatives 1,2,3 below applies)
take one of the following actions with probabilities $\lambda_1(t), \lambda_2(t), \lambda_3(t)$, respectively:
1. if (Φ contains a positive unit clause)
choose a random positive unit clause l
set l to TRUE
if this creates a contradiction FAIL
else
set a random variable to TRUE
2. if (Φ contains a negative unit clause)
choose a random negative unit clause $\bar{l}$
set l to FALSE
if this creates a contradiction FAIL
else
set a random variable to FALSE
3. if (Φ contains a clause of length ≥ 3)
choose a random clause C of maximal length
choose a random literal l of C
set l to FALSE

else
run the corresponding part of LARGEST-CLAUSE.

Fig. 4. The "lazy-server" version of algorithm LARGEST-CLAUSE

the dynamics of the four-tuple $(P(t), N(t), C_2(t), C_3(t))$:

$$\begin{cases} C_3(t+1) = C_3(t) - U_3(t) - \Delta_3(t), \\ C_2(t+1) = C_2(t) - \Delta_2(t) + \Delta_{3,2}(t), \\ P(t+1) = P(t) - U_P(t) - \Delta_{1,P}(t) + \Delta_{2,P}(t), \\ N(t+1) = N(t) - U_N(t) - \Delta_{1,N}(t) + \Delta_{2,N}(t) + \Delta_{3,N}(t), \end{cases} \tag{15}$$

where

$$\begin{cases} \Delta_3(t) \stackrel{d}{=} Bin\Big(C_3(t) - U_3(t), \frac{3}{n-t}\Big). \\ \Delta_2(t) = \Delta_{2,N}(t) + \Delta_{2,P}(t) \stackrel{d}{=} Bin\Big(C_2(t), \frac{2}{n-t}\Big). \\ \Delta_{3,2}(t) \stackrel{d}{=} U_3(t) + (U_N(t) + U_3(t)) \cdot Bin\Big(C_3(t) - U_3(t), \frac{3}{n-t}\Big) \\ \Delta_{3,N}(t) \stackrel{d}{=} 2U_P(t) \cdot Bin\Big(C_3(t), \frac{3}{n-t}\Big) \\ \Delta_{2,P}(t) \stackrel{d}{=} (U_N(t) + U_3(t)) \cdot Bin\Big(C_2(t), \frac{2}{n-t}\Big) \\ \Delta_{2,N}(t) \stackrel{d}{=} U_P(t) \cdot Bin\Big(C_2(t), \frac{2}{n-t}\Big) \\ \Delta_{1,P}(t) \stackrel{d}{=} Bin\Big(P(t) - U_P(t), \frac{1}{n-t}\Big) \\ \Delta_{1,N}(t) \stackrel{d}{=} Bin\Big(N(t) - U_N(t), \frac{1}{n-t}\Big) \end{cases} \tag{16}$$

By an analysis very similar to that of algorithm for random k-SAT (see e.g. [Ach01]), we derive the following system of equations that describe the median trajectory path of Algorithm LAZY LARGEST-CLAUSE:

$$\begin{cases} \dot{c_3}\ (t) = -\lambda_3(t) - \frac{3c_3(t)}{(1-t)}. \\ \dot{c_2}\ (t) = -\frac{2c_2(t)}{(1-t)} + \frac{3c_3(t)}{(1-t)} \cdot (\lambda_2(t) + \lambda_3(t)), \end{cases} \tag{17}$$

with initial conditions $(c_2(0), c_3(0)) = (0, r)$.

We will make the simplest choice

$$\lambda_1(t) = \lambda_2(t) = \lambda_3(t) = 1/3. \tag{18}$$

Differential equations (17) describe the dynamics of algorithm LARGEST-CLAUSE only for $t \in [t_3, t_2)$, where $t_3 = 0$ and $t_2 \in (0, 1)$ is the smallest solution of equation $c_3(t) = 0$.

Simple computations lead us to formulas:

$$\begin{cases} c_3(t) = (r+\frac{1}{6})(1-t)^3 - \frac{1-t}{6}, \\ c_2(t) = \frac{(1-t)^2}{3} - \frac{(1-t)}{3} + 2(r+\frac{1}{6})t(1-t)^2, \end{cases} \tag{19}$$

which describe the dynamics of algorithm LARGEST-CLAUSE in range $0 \leq t < t_2 = 1 - \frac{1}{\sqrt{6r+1}}$.

The flow into positive unit clauses is

$$\begin{aligned} F_2^P(t) &:= \frac{2}{3}\cdot\frac{2c_2(t)}{1-t} + \frac{1}{3}\cdot\frac{2\cdot 3c_3(t)}{1-t} = \\ &= \frac{4}{3}\Big[\frac{(1-t)^2}{3} - \frac{(1-t)}{3} + 2(r+\frac{1}{6})t(1-t)\Big] + 2(r+\frac{1}{6})(1-t)^2 - \frac{1}{3}. \end{aligned}$$

$$(F_2^P)'(t) = \frac{8r(1-2t)}{3} - 4(r+\frac{1}{6})(1-t) = \Big(\frac{4r}{3} - \frac{2}{3}\Big)(1-t) - \frac{8r}{3} < 0,$$

so $F_2^P(t)$ has a maximum at 0, equal to $2r$. For $r < 1/6$ this is less than $1/3$, so it is balanced by being given the opportunity (with probability $1/3$) to consume a positive unit clause, if any.

The average flow into negative unit clauses is

$$F_2^N(t) = \frac{1}{3}\cdot\frac{2c_2(t)}{1-t} = \frac{2}{3}\cdot\Big[\frac{(1-t)}{3} - \frac{1}{3} + 2(r+\frac{1}{6})t(1-t).\Big] = \frac{2t}{9}\Big[(6r+1)(1-t) - 1\Big].$$

The maximum of $F_2^N(t)$ is attained at $t = \frac{3r}{6r+1}$, which is in the interval (t_3, t_2) for $r > 0$, and is equal to $\frac{2r^2}{6r+1} = \frac{r}{3}(1 - \frac{1}{6r+1})$, which is definitely less than $\frac{1}{3}$ for $r < 1/6$.

The conclusion is that for $r < 1/6$ with probability $1 - o(1)$ both flows into positive and negative unit clauses can be handled by the lazy server with choice $\lambda_1 = \lambda_2 = \lambda_3 = 1/3$ without creating contradictory clauses.

Around stage $t_2 n \pm o(n)$ clauses of length three and one run out. We are left with a system of $(c_2(t_2) + o(1))n$ 1-in-2 clauses in the remaining $\overline{n} = (1-t_2)n$ variables. Consider graph G corresponding to these equations, where for every equation $x \oplus y = 1$ we add edge (x, y) to G.

By the uniformity lemma *G can be seen as an Erdős-Renyi random graph* $G(\overline{n}, \frac{\mu}{\overline{n}})$, with probability coefficient

$$\mu = 2c_2(t_2)/(1-t_2) = 3F_2(t_2).$$

Our maximum computation shows that for $r \in (0, 1/6)$, $3F_2(t_2) < 1$. Thus G is a subcritical random graph, whose connected components are w.h.p. of size $O(\log n)$. With constant probability (depending only on μ), G is a bipartite graph. In this situation giving a value to an arbitrary node uniquely determines the values of variables in the connected component.

We create two assignments A and B as follows:

(1) On variables x set by algorithm LARGEST-CLAUSE, $A(x) = B(x)$, equal to the value given by the algorithm.
(2) On variables in graph G A and B take opposite values. This can be accomplished by giving A, B different values on a set of fixed variables, one in each connected component of G.

When graph G is bipartite A and B are satisfying assignments. When the connected components of G are of size $O(\log n)$ we can create a path from A to B consisting satisfying assignments by consecutively flipping values of variables on which A and B are different, one connected component at a time. The overlap of A and B is equal to $1 - \frac{1}{\sqrt{6r+1}}$.

It follows that for any $q \in (0, 1)$, the q-overlap Exact Cover is satisfiable w.h.p. for $\frac{1}{6r+1} > (1-q)^2$, i.e. for $r < r_{lb}(q)$. $\square$

6 Remarks

The condition $r < 1/6$ in Theorem 1 has an easy probabilistic interpretation: it is the location of the phase transition for the random 3-uniform hypergraph [SPS85]. In this range most connected components are small and tree-like or unicyclic, so the space of variables breaks down in independent clusters of size $O(\log n)$. Thus we should expect that all overlaps in some range $(\lambda, 1)$ are satisfied with probability $1 - o(1)$, which is exactly what happens, according to Theorem 1, for $\lambda = 1 - \frac{1}{\sqrt{2}}$.

The relative weakness of this bound comes from our suboptimal choice of parameters $\lambda_1(t), \lambda_2(t), \lambda_3(t)$. For instance, the bound $r < 1/6$ comes entirely from handling positive unit clauses, while we have no problem satisfying negative ones, since the flow $F_2^N(t)$ always stays below one. This suggests that we are disproportionately often taking care of negative unit literals.

In what follows we sketch an approach for a better choice of these parameters. We were not able to explicitly calculate $\lambda_1(t), \lambda_2(t), \lambda_3(t)$, so we are unable to offer an improved analysis of the LAZY LARGEST-CLAUSE algorithm.

First, aiming to balance flows into $P(t)$, $N(t)$ we would like to satisfy

$$\frac{\lambda_1(t)\cdot(6c_3(t)+2c_2(t))}{1-t} = \frac{(\lambda_2(t)+\lambda_3(t))\cdot 2c_2(t))}{1-t},$$

i.e.

$$\lambda_1(t)\cdot(6c_3(t)+2c_2(t)) = (1-\lambda_1(t))\cdot 2c_2(t).$$

In other words we want

$$\lambda_1(t) = \frac{c_2(t)}{3c_3(t)+2c_2(t)}. \tag{20}$$

We give the algorithm the chance to satisfy accumulating positive unit clauses at each step with probability $\lambda_1(t)$.

The average flow into negative unit clauses is

$$F_2(t) := \lambda_3(t)\cdot\frac{6c_3(t)}{1-t} + \lambda_1(t)\cdot\frac{2c_2(t)}{1-t}.$$

To keep the buffer from overflowing we have to have $\lambda_2(t) > F_2(t)$. By replacing $\lambda_1(t), \lambda_3(t)$ in the expression of $F_2(t)$ we get the requirement

$$\lambda_2(t) > \left(1 - \frac{c_2(t)}{3c_3(t)+2c_2(t)} - \lambda_2(t)\right)\cdot\frac{6c_3(t)}{1-t} + \frac{c_2(t)}{3c_3(t)+2c_2(t)}\cdot\frac{2c_2(t)}{1-t}.$$

We will choose

$$\lambda_2(t) = \frac{\frac{3c_3(t)+c_2(t)}{3c_3(t)+2c_2(t)}\cdot\frac{6c_3(t)}{1-t} + \frac{c_2(t)}{3c_3(t)+2c_2(t)}\cdot\frac{2c_2(t)}{1-t}}{1+\frac{6c_3(t)}{1-t}} + \epsilon.$$

or, after some simple manipulations,

$$\lambda_2(t) = \frac{2c_2^2(t)+6c_2(t)c_3(t)+18c_3^2(t)}{[3c_3(t)+c_2(t)][6c_3(t)+1-t]} + \epsilon.$$

Finally, $\lambda_3(t)$ is determined from relation $\lambda_1(t)+\lambda_2(t)+\lambda_3(t)=1$. In turn, r should be upper bounded by requiring that $0 \le \lambda_1(t), \lambda_2(t), \lambda_3(t) \le 1$. This, however, requires computing $c_2(t)$, $c_3(t)$, a task that is complicated by the fact that parameters $\lambda_1(t), \lambda_2(t), \lambda_3(t)$ *depend* on these quantities, but also *influence* their evolution .

It is an open problem if this approach can be completed to a full analysis.

Conclusions and Acknowledgments

The obvious question raised by this work is to improve our bounds enough to display the discontinuity of overlap distribution, a property of k-Exact Cover we believe true. Note that there are obvious candidate approaches to improving our bounds: first, the upper bound could be improved by trying a rigorous version of the (heuristic) upper bound approach of Knysh et al. [KSM04]. On the other hand, the lower bound could be improved by finding explicit expressions for the parameters in (and explicitly analyzing) the LAZY LARGEST-CLAUSE algorithm, along the lines described in the previous section. Neither one of these two approaches looks particularly tractable, though.

This work has been supported by a Marie Curie International Reintegration Grant within the 6th European Community Framework Program.

References

[Ach01] D. Achlioptas. Lower bounds for random 3-SAT via differential equations. *Theoretical Computer Science*, 265:159–185, 2001.

[DMMZ08] H. Daudé, M. Mézard, T. Mora, and R. Zecchina. Pairs of SAT assignments and clustering in random boolean formulae. *Theoretical Computer Science*, (doi:10.1016/j.tcs.2008.01.005), 2008.

[HW05] A. Hartmann and M. Weigt. *Phase transitions in combinatorial optimization problems.* Wiley-VCH, 2005.

[Ist07] G. Istrate. Satisfiability of random Boolean CSP: Clusters and overlaps. *Journal of Universal Computer Science*, 13(11):1655–1670, 2007.

[KM05] Vamsi Kalapala and Cris Moore. The phase transition in exact cover. Technical Report cs/0508037, arXiv.org, 2005.

[KSM04] S. Knysh, V. N. Smelyanskiy, and R. D. Morris. Approximating satisfiability transition by suppressing fluctuations. Technical Report cond-mat/0403416, arXiv.org, 2004.

[MMR+07] E. Maneva, T. Meltzer, J. Raymond, A. Sportiello, and L. Zdeborová. A hike in the phases of the 1-in-3 satisfiability problem. In J.P. Bouchaud, M. Mézard, and J. Dalibard, editors, *Lecture Notes of the Les Houches Summer School 2006*, pages 491–498. Elsevier, 2007.

[MMZ05] M. Mézard, T. Mora, and R. Zecchina. Clustering of solutions in the random satisfiability problem. *Physical Review Letters*, 94(197205), 2005.

[PIM06] A. Percus, G. Istrate, and C. Moore, editors. *Computational Complexity and Statistical Physics*. Oxford University Press, 2006.

[RSZ07] J. Raymond, A. Sportiello, and L. Zdeborová. The phase diagram of random 1-in-3 satisfiability. *Phys. Rev. E*, 76(011101), 2007.

[SPS85] J. Schmidt-Pruznan and D. Shamir. Component structure in the evolution of random hypergraphs. *Combinatorica*, 5:81–94, 1985.