

BELIEF PROPAGATION GUIDED DECIMATION ON RANDOM k -XORSAT*

ARNAB CHATTERJEE, AMIN COJA-OGHLAN, MIHYUN KANG, LENA KRIEG, MAURICE ROLVIEN, GREGORY B. SORKIN

ABSTRACT. We analyse the performance of *Belief Propagation Guided Decimation*, a physics-inspired message passing algorithm, on the random k -XORSAT problem. Specifically, we derive an explicit threshold up to which the algorithm succeeds with a strictly positive probability $\Omega(1)$ that we compute explicitly, but beyond which the algorithm with high probability fails to find a satisfying assignment. In addition, we analyse a thought experiment called the *decimation process* for which we identify a (non-)reconstruction and a condensation phase transition. The main results of the present work confirm physics predictions from [Ricci-Tersenghi and Semerjian: J. Stat. Mech. 2009] that link the phase transitions of the decimation process with the performance of the algorithm, and improve over partial results from a recent article [Yung: Proc. ICALP 2024].

MSc: 60B20, 68W20

1. INTRODUCTION AND RESULTS

1.1. Background and motivation. The random k -XORSAT problem shares many characteristics of other intensely studied random constraint satisfaction problems ('CSPs') such as random k -SAT. For instance, as the clause/variable density increases, random k -XORSAT possesses a sharp satisfiability threshold preceded by a reconstruction or 'shattering' phase transition that affects the geometry of the set of solutions [2, 12, 17, 24]. As in random k -SAT, these transitions appear to significantly impact the performance of certain classes of algorithms [6, 16]. At the same time, random k -XORSAT is more amenable to mathematical analysis than, say, random k -SAT. This is because the XOR operation is equivalent to addition modulo two, which is why a k -XORSAT instance translates into a linear system over \mathbb{F}_2 . In effect, k -XORSAT can be solved in polynomial time by means of Gaussian elimination. In addition, the algebraic nature of the problem induces strong symmetry properties that simplify its study [3].

Because of its similarities with other random CSPs combined with said relative amenability, random k -XORSAT provides an instructive benchmark. This was noticed not only in combinatorics, but also in the statistical physics community, which has been contributing intriguing 'predictions' on random CSPs since the early 2000s [19, 22]. Among other things, physicists have proposed a message passing algorithm called *Belief Propagation Guided Decimation* ('BPGD') that, according to computer experiments, performs impressively on various random CSPs [21]. Furthermore, Ricci-Tersenghi and Semerjian [25] put forward a heuristic analysis of BPGD on random k -SAT and k -XORSAT. Their heuristic analysis proceeds by way of a thought experiment based on an idealised version of the algorithm. We call this thought experiment the *decimation process*. Based on physics methods Ricci-Tersenghi and Semerjian surmise that the decimation process undergoes two phase transitions, specifically a reconstruction and a condensation transition. A key prediction of Ricci-Tersenghi and Semerjian is that these phase transitions are directly linked to the performance of the BPGD algorithm. Due to the linear algebra-induced symmetry properties, in the case of random k -XORSAT all of these conjectures come as elegant analytical expressions.

The aim of this paper is to verify the predictions from [25] on random k -XORSAT mathematically. Specifically, our aim is to rigorously analyse the BPGD algorithm on random k -XORSAT, and to establish the link between its performance and the phase transitions of the decimation process. A first step towards a rigorous analysis of BPGD on random k -XORSAT was undertaken in a recent contribution by Yung [27]. However, Yung's analysis turns out to be not tight. Specifically, apart from requiring spurious lower bounds on the clause length k , Yung's results do not quite establish the precise connection between the decimation process and the performance of BPGD. One reason for this is that [27] relies on 'annealed' techniques, i.e., essentially moment computations. Here we instead harness 'quenched' arguments that were partly developed in prior work on the rank of random matrices over finite fields [3, 8].

* An Extended abstract appeared in the proceedings of ICALP 2025

Throughout we let $k \geq 3$ and $n \geq k$ be integers and $d > 0$ a positive real. Let $\mathbf{m} \stackrel{\text{dist}}{=} \text{Po}(dn/k)$ and let $F = F(n, d, k)$ be a random k -XORSAT formula¹ with variables x_1, \dots, x_n and \mathbf{m} random clauses of length k .

To be precise, every clause of F is an XOR of precisely k distinct variables, each of which may or may not come with a negation sign. The \mathbf{m} clauses are drawn uniformly and independently out of the set of all $2^k \binom{n}{k}$ possibilities. Thus, d equals the average number of clauses that a given variable x_i appears in. An event \mathcal{E} occurs *with high probability* ('w.h.p.') if $\lim_{n \rightarrow \infty} \mathbb{P}[\mathcal{E}] = 1$. We always keep d, k fixed as $n \rightarrow \infty$.

1.2. Belief Propagation Guided Decimation. The first result vindicates the predictions from [25] concerning the success probability of BPGD algorithm. BPGD sets its ambitions higher than merely finding a solution to the k -XORSAT instance F : the algorithm attempts to sample a solution uniformly at random. To this end BPGD assigns values to the variables x_1, \dots, x_n of F one after the other. In order to assign the next variable the algorithm attempts to compute the marginal probability that the variable is set to 'true' under a random solution to the k -XORSAT instance, given all previous assignments. More precisely, suppose BPGD has assigned values to the variables x_1, \dots, x_t already. Write $\sigma_{\text{BP}}(x_1), \dots, \sigma_{\text{BP}}(x_t) \in \{0, 1\}$ for their values, with 1 representing 'true' and 0 'false'. Further, let $F_{\text{BP},t}$ be the simplified formula obtained by substituting $\sigma_{\text{BP}}(x_1), \dots, \sigma_{\text{BP}}(x_t)$ for x_1, \dots, x_t . We drop any clauses from $F_{\text{BP},t}$ that contain variables from $\{x_1, \dots, x_t\}$ only, deeming any such clauses satisfied. Thus, $F_{\text{BP},t}$ is a XORSAT formula with variables x_{t+1}, \dots, x_n . Its clauses contain at least one and at most k variables, as well as possibly a constant (the XOR of the values substituted in for x_1, \dots, x_t).

Let $\sigma_{F_{\text{BP},t}}$ be a uniformly random solution of the XORSAT formula $F_{\text{BP},t}$, assuming that $F_{\text{BP},t}$ remains satisfiable. Then BPGD aims to compute the marginal probability $\mathbb{P}[\sigma_{F_{\text{BP},t}}(x_{t+1}) = 1 \mid F_{\text{BP},t}]$ that a random satisfying assignment of $F_{\text{BP},t}$ sets x_{t+1} to true. This is where Belief Propagation ('BP') comes in. An efficient message passing heuristic for computing precisely such marginals, BP returns an 'approximation' $\mu_{F_{\text{BP},t}}$ of $\mathbb{P}[\sigma_{F_{\text{BP},t}}(x_{t+1}) = 1 \mid F_{\text{BP},t}]$. We will recap the mechanics of BP in Section 2.2 (the value $\mu_{F_{\text{BP},t}}$ is defined precisely in (2.9)). Having computed the BP 'approximation', BPGD proceeds to assign x_{t+1} the value 'true' with probability $\mu_{F_{\text{BP},t}}$, otherwise sets x_{t+1} to 'false', then moves on to the next variable. The pseudocode is displayed as Algorithm 1.

Data: a random k -XORSAT formula F with variables x_1, \dots, x_n conditioned on being satisfiable

```

1 for  $t = 0, \dots, n-1$  do
2   compute the BP approximation  $\mu_{F_{\text{BP},t}}$ ;
3   set  $\sigma_{\text{BP}}(x_{t+1}) = \begin{cases} 1 & \text{with probability } \mu_{F_{\text{BP},t}} \\ 0 & \text{with probability } 1 - \mu_{F_{\text{BP},t}} \end{cases}$ ;
4 return  $\sigma_{\text{BP}}$ ;
```

Algorithm 1: The BPGD algorithm.

Let us pause for a few remarks. First, if the BP approximations are exact, i.e., if $F_{\text{BP},t}$ is satisfiable and $\mu_{F_{\text{BP},t}} = \mathbb{P}[\sigma_{F_{\text{BP},t}}(x_{t+1}) = 1 \mid F_{\text{BP},t}]$ for all t , then Bayes' formula shows that BPGD outputs a uniformly random solution of F . However, there is no universal guarantee that BP returns the correct marginals. Accordingly, the crux of analysing BPGD is precisely to figure out whether this is the case. Indeed, the heuristic work of [25] ties the accuracy of BP to a phase transition of the decimation process thought experiment, to be reviewed momentarily.

Second, the strategy behind the BPGD algorithm, particularly the message passing heuristic for 'approximating' the marginals, generalises well beyond k -XORSAT. For instance, the approach applies to k -SAT verbatim. That said, due to the algebraic nature of the XOR operation, BPGD is *far* easier to analyse on k -XORSAT. In fact, in XORSAT the marginal probabilities are guaranteed to be half-integral as seen in Fact 2.3, i.e.,

$$\mathbb{P}[\sigma_{F_{\text{BP},t}}(x_{t+1}) = 1 \mid F_{\text{BP},t}] \in \{0, 1/2, 1\}. \quad (1.1)$$

As a consequence, on XORSAT the BPGD algorithm effectively reduces to a purely combinatorial algorithm called Unit Clause Propagation [19, 25] as per Proposition 6.1, a fact that we will exploit extensively (see Section 6).

¹Two random variables X, Y are equal in distribution $X \stackrel{\text{dist}}{=} Y$ if they have the same distribution functions. Thus, \mathbf{m} follows a Poisson distribution with mean dn/k .

1.3. A tight analysis of BPGD. In order to state the main results we need to introduce a few threshold values. To this end, given d, k and an additional real parameter $\lambda \geq 0$ that depends on the time t , consider the functions²

$$\phi_{d,k,\lambda} : [0, 1] \rightarrow [0, 1], \quad z \mapsto 1 - \exp(-\lambda - dz^{k-1}), \quad (1.2)$$

$$\Phi_{d,k,\lambda} : [0, 1] \rightarrow \mathbb{R}, \quad z \mapsto \exp(-\lambda - dz^{k-1}) - \frac{d(k-1)}{k} z^k + dz^{k-1} - \frac{d}{k}. \quad (1.3)$$

Let $\alpha_*(\lambda) = \alpha_*(d, k, \lambda) \in [0, 1]$ be the smallest and $\alpha^*(\lambda) = \alpha^*(d, k, \lambda) \geq \alpha_*(d, k, \lambda) \in [0, 1]$ the largest fixed point of $\phi_{d,k,\lambda}$. Figure 1 visualises $\Phi(z)$ for different values of $\theta \sim t/n$. Further, define

$$d_{\min}(k) = \left(\frac{k-1}{k-2}\right)^{k-2}, \quad d_{\text{core}}(k) = \sup\{d > 0 : \alpha^*(0) = 0\}, \quad d_{\text{sat}}(k) = \sup\{d > 0 : \Phi_{d,k,0}(\alpha^*(0)) \leq \Phi_{d,k,0}(0)\}. \quad (1.4)$$

The value $d_{\text{sat}}(k)$ is the random k -XORSAT satisfiability threshold [3, 12, 24]. Thus, for $d < d_{\text{sat}}(k)$ the random k -XORSAT formula F possesses satisfying assignments w.h.p., while F is unsatisfiable for $d > d_{\text{sat}}(k)$ w.h.p. Furthermore, $d_{\text{core}}(k)$ equals the threshold for the emergence of a giant 2-core within the k -uniform hypergraph induced by F [3, 23]. This implies that for $d < d_{\text{core}}(k)$ the set of solutions of F is contiguous in a certain well-defined way, while for $d_{\text{core}}(k) < d < d_{\text{sat}}(k)$ the set of solutions shatters into an exponential number of well-separated clusters [16, 19]. Moreover, a simple linear time algorithm is known to find a solution w.h.p. for $d < d_{\text{core}}(k)$ [16]. The relevance of $d_{\min}(k)$ will emerge in Theorem 1.1. A bit of calculus reveals that

$$0 < d_{\min}(k) < d_{\text{core}}(k) < d_{\text{sat}}(k) < k. \quad (1.5)$$

The following theorem determines the precise clause-to-variable densities where BPGD succeeds/fails. To be precise, in the ‘successful’ regime BPGD does not actually succeed with *high* probability, but with an explicit probability strictly between zero and one, which is displayed in Figure 2 for $k = 3, 4, 5$.

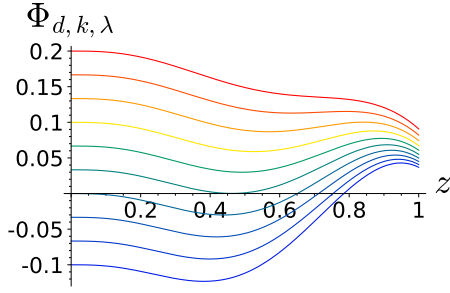


FIGURE 1. $\Phi_{d,k,\lambda}$ for $k = 3$ and $d = 2.4$, for λ from 0 to 0.3 (maximum at $z = 0$) and from 0.4 to 0.9

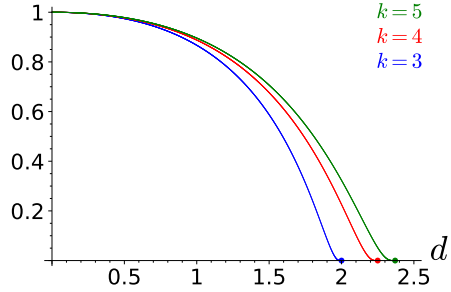


FIGURE 2. Success probability of BPGD for $0 < d < d_{\min}(k)$ and various k .

Theorem 1.1. *Let $k \geq 3$.*

(i) *If $d < d_{\min}(k)$, then*

$$\lim_{n \rightarrow \infty} \mathbb{P}[\text{BPGD}(F) \text{ finds a satisfying assignment}] = \exp\left(-\frac{d^2(k-1)^2}{4} \int_0^1 \frac{z^{2k-4}(1-z)}{1-d(k-1)z^{k-2}(1-z)} dz\right). \quad (1.6)$$

(ii) *If $d_{\min}(k) < d < d_{\text{sat}}(k)$, then*

$$\mathbb{P}[\text{BPGD}(F) \text{ finds a satisfying assignment}] = o(1).$$

Theorem 1.1 vindicates the predictions from Ricci-Tersenghi and Semerjian [25, Section 4] as to the performance of BPGD, and improves over the results from Yung [27]. Specifically, Theorem 1.1 (i) verifies the formula for the success probability from [25, Eq. (38)]. Combinatorially, the formula (1.6) results from the possible presence of bounded length cycles (so-called toxic cycles) that may cause the algorithm to run into contradictions. This complements Yung’s prior work, that has no positive result on the performance of BPGD. Moreover, Yung’s negative

²The function $\Phi_{d,k,\lambda}$ is known in physics parlance as the ‘Bethe free entropy’ [8, 19]. The stationary points of $\Phi_{d,k,\lambda}$ coincide with the fixed points of $\phi_{d,k,\lambda}$, as we will verify in Section 2.1.

results [27, Theorems 2–3] only apply to $k \geq 9$ and to $d > d_{\text{core}}(k)$, while Theorem 1.1 (ii) covers all $k \geq 3$ and kicks in at the correct threshold $d_{\min}(k) < d_{\text{core}}(k)$ predicted in [25].

1.4. The decimation process. In addition to the BPGD algorithm itself, the heuristic work [25] considers an idealised version of the algorithm, the *decimation process*. This thought experiment highlights the conceptual reasons behind the success/failure of BPGD. Just like BPGD, the decimation process assigns values to variables one after the other for good. But instead of the BP ‘approximations’ the decimation process uses the *actual* marginals given its previous decisions. To be precise, suppose that the input formula F is satisfiable and that variables x_1, \dots, x_t have already been assigned values $\sigma_{\text{DC}}(x_1), \dots, \sigma_{\text{DC}}(x_t)$ in the previous iterations. Obtain $F_{\text{DC},t}$ by substituting the values $\sigma_{\text{DC}}(x_1), \dots, \sigma_{\text{DC}}(x_t)$ for x_1, \dots, x_t and dropping any clauses that do not contain any of x_{t+1}, \dots, x_n . Thus, $F_{\text{DC},t}$ is a XORSAT formula with variables x_{t+1}, \dots, x_n . Let $\sigma_{F_{\text{DC},t}}$ be a random satisfying assignment of $F_{\text{DC},t}$. Then the decimation process sets x_{t+1} according to the true marginal $\mathbb{P}[\sigma_{F_{\text{DC},t}}(x_{t+1}) = 1 \mid F_{\text{DC},t}]$, thus ultimately returning a uniformly random satisfying assignment of F .

Data: a random k -XORSAT formula F , conditioned on being satisfiable

```

1 for  $t = 0, \dots, n-1$  do
2   compute  $\pi_{F_{\text{DC},t}} = \mathbb{P}[\sigma_{F_{\text{DC},t}}(x_{t+1}) = 1 \mid F_{\text{DC},t}]$ ;
3   set  $\sigma_{\text{DC}}(x_t) = \begin{cases} 1 & \text{with probability } \pi_{F_{\text{DC},t}} \\ 0 & \text{with probability } 1 - \pi_{F_{\text{DC},t}} \end{cases}$ ;
4 return  $\sigma_{\text{DC}}$ ;
```

Algorithm 2: The decimation process.

Clearly, if indeed the BP ‘approximations’ are correct, then the decimation process and BPGD are identical. Thus, a key question is for what parameter regimes the two process coincide or diverge, respectively. As it turns out, this question is best answered by parametrize not only in terms of the average variable degree d , but also in terms of the ‘time’ parameter t of the decimation process.

1.5. Phase transitions of the decimation process. Ricci-Tersenghi and Semerjian heuristically identify several phase transitions in terms of d and t that the decimation process undergoes. We will confirm these predictions mathematically and investigate how they relate to the performance of BPGD.

The first set of relevant phase transitions concerns the so-called non-reconstruction property. Roughly speaking, non-reconstruction means that the marginal $\pi_{F_{\text{DC},t}} = \mathbb{P}[\sigma_{F_{\text{DC},t}}(x_{t+1}) = 1 \mid F_{\text{DC},t}]$ is determined by short-range rather than long-range effects. Since Belief Propagation is essentially a local algorithm, one might expect that the (non-)reconstruction phase transition coincides with the threshold up to which BPGD succeeds; cf. the discussions in [5, 17].

To define (non-)reconstruction precisely, we associate a bipartite graph $G(F_{\text{DC},t})$ with the formula $F_{\text{DC},t}$. The vertices of this graph are the variables and clauses of $F_{\text{DC},t}$. Each variable is adjacent to the clauses in which it appears. For a (variable or clause) vertex v of $G(F_{\text{DC},t})$ let ∂v be the set of neighbours of v in $G(F_{\text{DC},t})$. More generally, for an integer $\ell \geq 1$ let $\partial^\ell v$ be the set of vertices of $G(F_{\text{DC},t})$ at shortest path distance precisely ℓ from v . Following [17], we say that $F_{\text{DC},t}$ has the *non-reconstruction property* if

$$\lim_{\ell \rightarrow \infty} \limsup_{n \rightarrow \infty} \mathbb{E} \left[\left| \mathbb{P}[\sigma_{F_{\text{DC},t}}(x_{t+1}) = 1 \mid F_{\text{DC},t}, \{\sigma_{F_{\text{DC},t}}(y)\}_{y \in \partial^{2\ell} x_{t+1}}] - \mathbb{P}[\sigma_{F_{\text{DC},t}}(x_{t+1}) = 1 \mid F_{\text{DC},t}] \right| \mid F \text{ satisfiable} \right] = 0. \quad (1.7)$$

Conversely, $F_{\text{DC},t}$ has the *reconstruction property* if

$$\liminf_{\ell \rightarrow \infty} \liminf_{n \rightarrow \infty} \mathbb{E} \left[\left| \mathbb{P}[\sigma_{F_{\text{DC},t}}(x_{t+1}) = 1 \mid F_{\text{DC},t}, \{\sigma_{F_{\text{DC},t}}(y)\}_{y \in \partial^{2\ell} x_{t+1}}] - \mathbb{P}[\sigma_{F_{\text{DC},t}}(x_{t+1}) = 1 \mid F_{\text{DC},t}] \right| \mid F \text{ sat.} \right] > 0. \quad (1.8)$$

To parse (1.7), notice that in the left probability term we condition on both the outcome $F_{\text{DC},t}$ of the first t steps of the decimation process and on the values $\sigma_{F_{\text{DC},t}}(y)$ that the random solution $\sigma_{F_{\text{DC},t}}$ assigns to the variables y at distance exactly 2ℓ from x_{t+1} . By contrast, in the right probability term we only condition on $F_{\text{DC},t}$. Thus, the second probability term matches the probability $\pi_{F_{\text{DC},t}}$ from the decimation process. Hence, (1.7) compares the probability that a random solution sets x_{t+1} to one given the values $\sigma_{F_{\text{DC},t}}(y)$ of *all* variables y at distance 2ℓ from x_{t+1} with plain marginal probability that x_{t+1} is set to one. What (1.7) asks is that these two probabilities

be asymptotically equal in the limit of large ℓ , with high probability over the choice of F and the prior steps of the decimation process. Thus, so long as non-reconstruction holds ‘long-range effects’, meaning anything beyond distance 2ℓ for large enough but fixed ℓ , are negligible.

Confirming the predictions from [25], the following theorem identifies the precise regimes of d, t where (non-)reconstruction holds. To state the theorem, we need to know that for $d_{\min}(k) < d < d_{\text{sat}}(k)$ the polynomial $d(k-1)z^{k-2}(1-z) - 1$ has precisely two roots $0 < z_* = z_*(d, k) < z^* = z^*(d, k) < 1$; we are going to prove this as part of Proposition 2.2 below. Let

$$\lambda_* = \lambda_*(d, k) = -\log(1 - z_*) - \frac{z_*}{(k-1)(1-z_*)} > \lambda^* = \lambda^*(d, k) = \max \left\{ 0, -\log(1 - z^*) - \frac{z^*}{(k-1)(1-z^*)} \right\} \geq 0, \quad (1.9)$$

$$\theta_* = \theta_*(d, k) = 1 - \exp(-\lambda_*) > \theta^* = \theta^*(d, k) = 1 - \exp(-\lambda^*). \quad (1.10)$$

Additionally, let $\lambda_{\text{cond}}(d, k)$ be the solution to the ODE

$$\frac{\partial \lambda_{\text{cond}}(d, k)}{\partial d} = -\frac{\alpha^*(\lambda_{\text{cond}}(d, k))^k - \alpha_*(\lambda_{\text{cond}}(d, k))^k}{k(\alpha^*(\lambda_{\text{cond}}(d, k)) - \alpha_*(\lambda_{\text{cond}}(d, k)))}, \quad \lambda_{\text{cond}}(d_{\text{sat}}(k), k) = 0 \quad (1.11)$$

on the interval $(d_{\min}, d_{\text{sat}}]$ and set $\theta_{\text{cond}} = \theta_{\text{cond}}(d, k) = 1 - \exp(-\lambda_{\text{cond}}(d, k))$. Note that

$$\theta^* < \theta_{\text{cond}} < \theta_*.$$

Theorem 1.2. *Let $k \geq 3$ and let $0 \leq t = t(n) \leq n$ be a sequence such that $\lim_{n \rightarrow \infty} t/n = \theta \in (0, 1)$.*

- (i) *If $d < d_{\min}(k)$, then $F_{\text{DC}, t}$ has the non-reconstruction property w.h.p.*
- (ii) *If $d_{\min}(k) < d < d_{\text{sat}}(k)$ and $\theta < \theta^*$ or $\theta > \theta_{\text{cond}}$, then $F_{\text{DC}, t}$ has the non-reconstruction property w.h.p.*
- (iii) *If $d_{\min}(k) < d < d_{\text{sat}}(k)$ and $\theta^* < \theta < \theta_{\text{cond}}$, then $F_{\text{DC}, t}$ has the reconstruction property w.h.p.*

Theorem 1.2 shows that $d_{\min}(k)$ marks the precise threshold of d up to which the decimation process $F_{\text{DC}, t}$ exhibits non-reconstruction for all $0 \leq t \leq n$ w.h.p. By contrast, for $d_{\min}(k) < d < d_{\text{sat}}(k)$ there is a regime of t where reconstruction occurs. In fact, as Proposition 2.2 shows, for $d > d_{\text{core}}(k)$ we have $\theta^* = 0$ and thus reconstruction holds even at $t = 0$, i.e., for the original, undecimated random formula F . Prior to the contribution [25], it had been suggested that this precise scenario (reconstruction on the original problem instance) is the stone on which BPGD stumbles [5]. In fact, Yung’s negative result kicks in at this precise threshold $d_{\text{core}}(k)$. However, Theorems 1.1 and 1.2 show that matters are more subtle. Specifically, for $d_{\min}(k) < d < d_{\text{core}}(k)$ reconstruction, even though absent in the initial formula F , occurs at a later ‘time’ $t > 0$ as decimation proceeds, which suffices to trip BPGD up. Also, remarkably, Theorem 1.2 shows that non-reconstruction is not ‘monotone’. The property holds for $\theta < \theta^*$ and then again for $\theta > \theta_{\text{cond}}$, but not on the interval $(\theta^*, \theta_{\text{cond}})$ as visualised in Figure 3.

But there is one more surprise. Namely, Theorem 1.2 (ii) might suggest that for $d_{\min}(k) < d < d_{\text{sat}}(k)$ Belief Propagation manages to compute the correct marginals for $t/n \sim \theta > \theta_{\text{cond}}$, as non-reconstruction kicks back in. But remarkably, this is not quite true. Despite the fact that non-reconstruction holds, BPGD goes astray because the algorithm starts its message passing process from a mistaken, oblivious initialisation. As a consequence, for $t/n \sim \theta \in (\theta_{\text{cond}}, \theta_*)$ the BP ‘approximations’ remain prone to error. To be precise, the following result identifies the precise ‘times’ where BP succeeds/fails. To state the result let $\mu_{F_{\text{DC}, t}}$ denote the BP ‘approximation’ of the true marginal $\pi_{F_{\text{DC}, t}}$ of variable x_{t+1} in the formula $F_{\text{DC}, t}$ created by the decimation process (see Section 2.2 for a reminder of the definition). Also recall that $\pi_{F_{\text{DC}, t}}$ denotes the correct marginal as used by the decimation process.

Theorem 1.3. *Let $k \geq 3$ and let $0 \leq t = t(n) \leq n$ be a sequence such that $\lim_{n \rightarrow \infty} t/n = \theta \in (0, 1)$.*

- (i) *If $0 < d < d_{\min}(k)$ then $\mu_{F_{\text{DC}, t}} = \pi_{F_{\text{DC}, t}}$ w.h.p.*
- (ii) *If $d_{\min}(k) < d < d_{\text{sat}}(k)$ and $\theta < \theta_{\text{cond}}$ or $\theta > \theta_*$, then $\mu_{F_{\text{DC}, t}} = \pi_{F_{\text{DC}, t}}$ w.h.p.*
- (iii) *If $d_{\min}(k) < d < d_{\text{sat}}(k)$ and $\theta_{\text{cond}} < \theta < \theta_*$, then $\mathbb{E}|\mu_{F_{\text{DC}, t}} - \pi_{F_{\text{DC}, t}}| = \Omega(1)$.*

The upshot of Theorems 1.2–1.3 is that the relation between the accuracy of BP and reconstruction is subtle. Everything goes well so long as $d < d_{\min}$ as non-reconstruction holds throughout and the BP approximations are correct. But if $d_{\min} < d < d_{\text{sat}}$ and $\theta^* < \theta < \theta_{\text{cond}}$, then Theorem 1.2 (iii) shows that reconstruction occurs. Nonetheless, Theorem 1.3 (ii) demonstrates that the BP approximations remain valid in this regime. By contrast, for $\theta_{\text{cond}} < \theta < \theta_*$ we have non-reconstruction by Theorem 1.2 (iii), but Theorem 1.3 (iii) shows that BP misses its mark with a non-vanishing probability. Finally, for $\theta > \theta_*$ everything is in order once again as BP regains its footing and non-reconstruction holds. Unfortunately BPGD is unlikely to reach this happy state because the algorithm is bound to make numerous mistakes at times $t/n \sim \theta \in (\theta_{\text{cond}}, \theta_*)$.

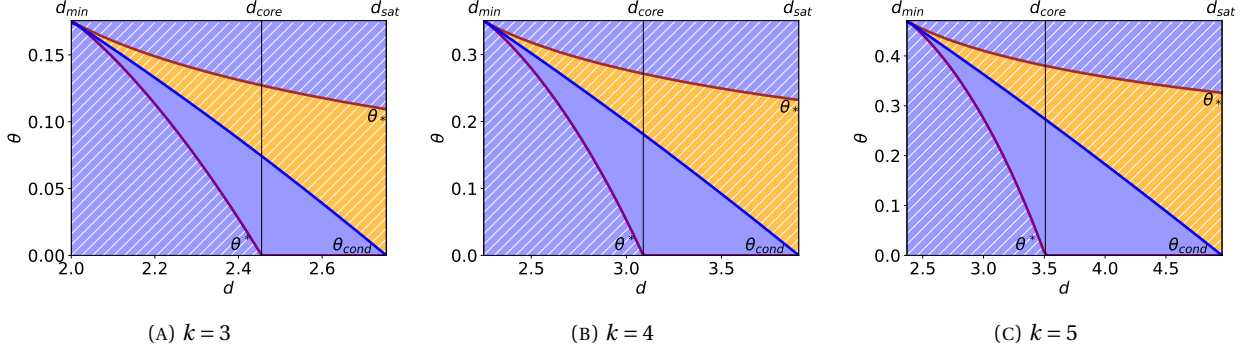


FIGURE 3. The phase diagrams for $k = 3, 4, 5$ with $d \in (d_{\min}, d_{\text{sat}})$ on the horizontal and θ on the vertical axis. The hatched area displays the regime $\theta < \theta^*$ and $\theta_{\text{cond}} < \theta$ where non-reconstruction holds. In the non-hatched area, where $\theta^* < \theta < \theta_{\text{cond}}$, we have reconstruction. Similarly, the blue area displays $\theta < \theta_{\text{cond}}$ and $\theta > \theta^*$ where BP is correct whereas in the orange area, BP is inaccurate.

Theorems 1.2 and 1.3 confirm the predictions from [25, Section 4]. To be precise, while θ_{cond} matches the predictions of Ricci-Tersenghi and Semerjian, the ODE formula (1.11) for the threshold, which is easy to evaluate numerically, does not appear in [25]. Instead of the ODE formulation, Ricci-Tersenghi and Semerjian define λ_{cond} as the (unique) $\lambda \geq 0$ such that $\Phi_{d,k,\lambda}(\alpha_*) = \Phi_{d,k,\lambda}(\alpha^*)$; Proposition 2.2 below shows that both are equivalent. Illustrating Theorems 1.2–1.3, Figure 3 displays the phase diagram in terms of d and $\theta \sim t/n$ for $k = 3, 4, 5$.

2. OVERVIEW

This section provides an overview of the proofs of Theorems 1.1–1.3. In the final paragraph we conclude with a discussion of further related work. *We assume throughout that $k \geq 3$ is an integer and that $0 < d < d_{\text{sat}}(k)$. Moreover, $t = t(n)$ denotes an integer sequence $0 \leq t(n) \leq n$ such that $\lim_{n \rightarrow \infty} t(n)/n = \theta \in (0, 1)$.*

2.1. Fixed points and thresholds. The first item on our agenda is to study the functions $\phi_{d,k,\lambda}, \Phi_{d,k,\lambda}$ from (1.2)–(1.3). Specifically, we are concerned with the maxima of $\Phi_{d,k,\lambda}$ and the fixed points of $\phi_{d,k,\lambda}$, the combinatorial relevance of which will emerge as we analyse BPGD and the decimation process. We begin by observing that the fixed points of $\phi_{d,k,\lambda}$ are precisely the stationary points of $\Phi_{d,k,\lambda}$.

Fact 2.1. *For any $d > 0, \lambda \geq 0$ the stationary points $z \in (0, 1)$ of $\Phi_{d,k,\lambda}$ coincide with the fixed points of $\phi_{d,k,\lambda}$ in $(0, 1)$. Furthermore, for a fixed point $z \in (0, 1)$ of $\phi_{d,k,\lambda}$ we have*

$$\Phi''_{d,k,\lambda}(z) \begin{cases} < 0 & \text{if } \phi'_{d,k,\lambda}(z) < 1, \\ = 0 & \text{if } \phi'_{d,k,\lambda}(z) = 1, \\ > 0 & \text{if } \phi'_{d,k,\lambda}(z) > 1. \end{cases} \quad (2.1)$$

Proof. **TOPROVE 0** □

We recall that $0 \leq \alpha_* = \alpha_*(d, k, \lambda) \leq \alpha^* = \alpha^*(d, k, \lambda) \leq 1$ are the smallest and the largest fixed point of $\phi_{d,k,\lambda}$ in $[0, 1]$, respectively. Fact 2.1 shows that $\Phi_{d,k,\lambda}$ attains its global maximum in $[0, 1]$ at α_* or α^* . Let

$$\alpha_{\max} = \alpha_{\max}(d, k, \lambda) \in \{\alpha_*, \alpha^*\}$$

be the maximiser of $\Phi_{d,k,\lambda}$; if $\Phi_{d,k,\lambda}(\alpha_*) = \Phi_{d,k,\lambda}(\alpha^*)$, set $\alpha_{\max} = \alpha^*$. The following proposition characterises the fixed points of $\phi_{d,k,\lambda}$ and the maximiser α_{\max} .

Proposition 2.2.

- (i) *If $d < d_{\min}(k)$, then for all $\lambda > 0$ we have $\alpha_*(d, k, \lambda) = \alpha^*(d, k, \lambda)$, the function $\lambda \in (0, \infty) \mapsto \alpha_*(d, k, \lambda) \in (0, 1)$ is analytic, and $\alpha_*(d, k, \lambda)$ is the unique stable fixed point of $\phi_{d,k,\lambda}$.*
- (ii) *If $d_{\min}(k) < d < d_{\text{sat}}(k)$, then the polynomial $d(k-1)z^{k-2}(1-z) - 1$ has precisely two roots $0 < z_* < z^* < 1$, the numbers λ_*, λ^* from (1.9) satisfy $0 \leq \lambda^* < \lambda_*$ and the following is true.*

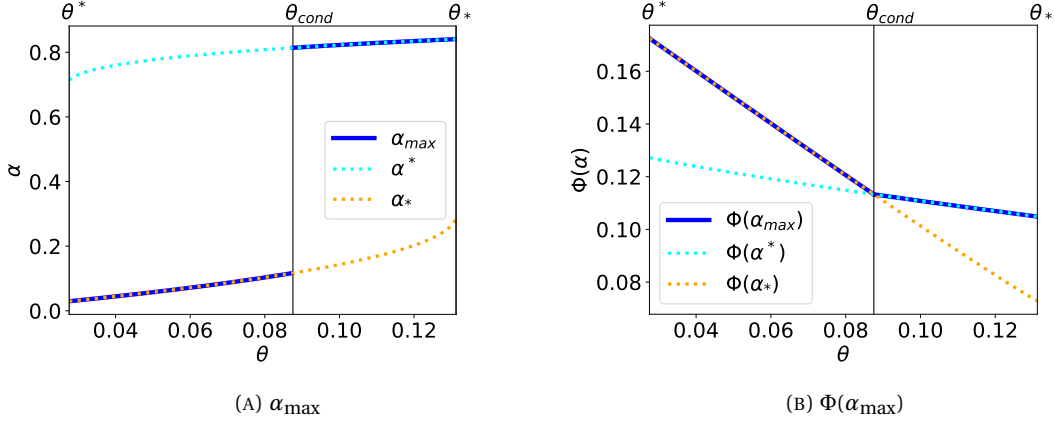


FIGURE 4. α_{\max} and $\Phi(\alpha_{\max}) = \Phi_{d,k,\lambda}(\alpha_{\max})$ for $d = 2.4$ and $k = 3$ from θ^* to θ_* .

- (a) If $\lambda < \lambda^*$ or $\lambda > \lambda_*$, then $\alpha_*(d, k, \lambda) = \alpha^*(d, k, \lambda) \in (0, 1)$ is the unique stable fixed point of $\phi_{d,k,\lambda}$.
- (b) If $\lambda^* < \lambda < \lambda_*$, then $0 < \alpha_*(d, k, \lambda) < \alpha^*(d, k, \lambda) < 1$ are the only stable fixed points of $\phi_{d,k,\lambda}$.
- (c) The functions $\lambda \in (0, \lambda_*) \mapsto \alpha_*(d, k, \lambda)$ and $\lambda \in (\lambda^*, \infty) \mapsto \alpha^*(d, k, \lambda)$ are analytic.
- (d) If $d_{\min}(k) < d < d_{\text{sat}}(k)$, then the solution λ_{cond} of (1.11) satisfies $\lambda^* < \lambda_{\text{cond}} = \lambda_{\text{cond}}(d) < \lambda_*$ and

$$\alpha_{\max}(d, k, \lambda) = \begin{cases} \alpha_*(d, k, \lambda) & \text{if } \lambda < \lambda_{\text{cond}}, \\ \alpha^*(d, k, \lambda) & \text{if } \lambda > \lambda_{\text{cond}}. \end{cases}$$

Furthermore, $\Phi_{d,k,\lambda}(\alpha^*(d, k, \lambda)) \neq \Phi_{d,k,\lambda}(\alpha_*(d, k, \lambda))$ unless $\lambda = \lambda_{\text{cond}}$. Thus, the function $\lambda \mapsto \alpha_{\max}(d, k, \lambda)$ is analytic on $(0, \lambda_{\text{cond}})$ and on $(\lambda_{\text{cond}}, \infty)$, but discontinuous at $\lambda = \lambda_{\text{cond}}$.

2.2. Belief Propagation. Having done our analytic homework, we proceed to recall how Belief Propagation computes the ‘approximations’ $\mu_{\text{BP},t}$ that the BPGD algorithm relies upon. We will see that due to the inherent symmetries of XORSAT the Belief Propagation computations simplify and boil down to a simpler message passing process called Warning Propagation. Subsequently we will explain the connection between Warning Propagation and the fixed points α_*, α^* of $\phi_{d,k,\lambda}$.

It is probably easiest to explain BP on a general XORSAT instance F with a set $V(F)$ of variables and a set $C(F)$ of clauses of lengths between one and k . As in Section 1.5 we consider the graph $G(F)$ induced by F , with vertex set $V(F) \cup C(F)$ and an edge xa between $x \in V(F)$ and $a \in C(F)$ iff a contains x . Let $\partial v = \partial_F v$ be the set of neighbours of $v \in V(F) \cup C(F)$. Additionally, given an assignment $\tau \in \{0, 1\}^{\partial a}$ of the variables that appear in a , we write $\tau \models a$ iff τ satisfies a .

With each variable/clause pair x, a such that $x \in \partial a$ Belief Propagation associates two sequences of ‘messages’ $(\mu_{F,x \rightarrow a,\ell})_{\ell \geq 0}, (\mu_{F,a \rightarrow x,\ell})_{\ell \geq 0}$ directed from x to a and from a to x , respectively. These messages are probability distributions on $\{0, 1\}$, i.e.,

$$\mu_{F,x \rightarrow a,\ell} = (\mu_{F,x \rightarrow a,\ell}(0), \mu_{F,x \rightarrow a,\ell}(1)), \mu_{F,a \rightarrow x,\ell} = (\mu_{F,a \rightarrow x,\ell}(0), \mu_{F,a \rightarrow x,\ell}(1)), \quad (2.2)$$

$$\mu_{F,x \rightarrow a,\ell}(0) + \mu_{F,x \rightarrow a,\ell}(1) = \mu_{F,a \rightarrow x,\ell}(0) + \mu_{F,a \rightarrow x,\ell}(1) = 1. \quad (2.3)$$

The initial messages are uniform, i.e.,

$$\mu_{F,x \rightarrow a,0}(s) = \mu_{F,a \rightarrow x,0}(s) = 1/2 \quad (s \in \{0, 1\}). \quad (2.4)$$

Further, the messages at step $\ell + 1$ are obtained from the messages at step ℓ via the *Belief Propagation equations*

$$\mu_{F,a \rightarrow x,\ell+1}(s) \propto \sum_{\tau \in \{0,1\}^{\partial a}} \mathbb{1}\{\tau_x = s, \tau \models a\} \prod_{y \in \partial a \setminus \{x\}} \mu_{F,y \rightarrow a,\ell}(\tau_y), \quad (2.5)$$

$$\mu_{F,x \rightarrow a,\ell+1}(s) \propto \prod_{b \in \partial x \setminus \{a\}} \mu_{F,b \rightarrow x,\ell}(s). \quad (2.6)$$

In (2.5)–(2.6) the \propto -symbol represents the normalisation required to ensure that the updated messages satisfy (2.3). In the case of (2.6) such a normalisation may be impossible because the expressions on the r.h.s. could vanish for

both $s = 0$ and $s = 1$. In this event we agree that

$$\mu_{F,x \rightarrow a, \ell+1}(s) = \begin{cases} \mu_{F,x \rightarrow a, \ell}(s) & \text{if } \mu_{F,x \rightarrow a, \ell}(s) \neq 1/2 \\ \mathbb{1}\{s=0\} & \text{otherwise} \end{cases} \quad (s \in \{0, 1\});$$

in other words, we retain the messages from the previous iteration unless its value was $1/2$, in which case we set $\mu_{F,x \rightarrow a, \ell+1}(0) = 1$. The same convention applies to $\mu_{F,a \rightarrow x, \ell+1}(s)$. Further, at any time t the BP messages render a heuristic ‘approximation’ of the marginal probability that a random solution to the formula F sets a variable x to $s \in \{0, 1\}$:

$$\mu_{F,x,\ell}(s) \propto \prod_{b \in \partial x} \mu_{F,b \rightarrow x, \ell}(s). \quad (2.7)$$

We set $\mu_{F,x,\ell}(0) = 1 - \mu_{F,x,\ell}(1) = 1$ if the normalisation in (2.7) fails, i.e., if $\sum_{s \in \{0,1\}} \prod_{b \in \partial x} \mu_{F,b \rightarrow x, \ell}(s) = 0$.

Fact 2.3. *The BP messages and marginals are half-integral for all t , i.e., for all $t \geq 0$ and $s \in \{0, 1\}$ we have*

$$\mu_{F,x \rightarrow a, \ell}(s), \mu_{F,a \rightarrow x, \ell}(s), \mu_{F,x,\ell}(s) \in \{0, 1/2, 1\}. \quad (2.8)$$

Furthermore, for all $\ell > 2 \sum_{a \in C(F)} |\partial a|$ we have $\mu_{F,x,\ell}(s) = \mu_{F,x,\ell+1}(s)$.

Proof. **TOPROVE 1** □

Finally, in light of Fact 2.3 it makes sense to define the approximations for BPGD by letting

$$\mu_{F_{BP},t} = \lim_{\ell \rightarrow \infty} \mu_{F_{BP},t,x_{t+1},\ell}(1), \quad \mu_{F_{DC},t} = \lim_{\ell \rightarrow \infty} \mu_{F_{DC},t,x_{t+1},\ell}(1). \quad (2.9)$$

2.3. Warning Propagation. Thanks to the half-integrality (2.8) of the messages, Belief Propagation is equivalent to a purely combinatorial message passing procedure called *Warning Propagation* (‘WP’) [19]. Similar as BP, WP also associates two message sequences $(\omega_{F,x \rightarrow a, \ell}, \omega_{F,a \rightarrow x, \ell})_{\ell \geq 0}$ with every adjacent variable/clause pair. The messages take one of three possible discrete values $\{f, u, n\}$ (‘frozen’, ‘uniform’, ‘null’). Essentially, n indicates that the value of a variable is determined by unit clause propagation. Moreover, f indicates that a variable is forced to take the value 0 once all variables in the 2-core of the hypergraph representation of the formula are set to 0. The remaining label u indicates that neither of the above applies. To trace the BP messages from Section 2.2 actually only the two values $\{n, u\}$ would be necessary. However, the third value f will prove useful in order to compare the BP approximations with the actual marginals. Perhaps unexpectedly given the all-uniform initialisation (2.4), we launch WP from all-frozen start values:

$$\omega_{F,x \rightarrow a, 0} = \omega_{F,a \rightarrow x, 0} = f \quad \text{for all } a, x. \quad (2.10)$$

Subsequently the messages get updated according to the rules

$$\omega_{F,a \rightarrow x, \ell+1} = \begin{cases} n & \text{if } \omega_{F,y \rightarrow a, \ell} = n \text{ for all } y \in \partial a \setminus \{x\}, \\ f & \text{if } \omega_{F,y \rightarrow a, \ell} \neq u \text{ for all } y \in \partial a \setminus \{x\} \text{ and } \omega_{F,y \rightarrow a, \ell} \neq n \text{ for at least one } y \in \partial a \setminus \{x\}, \\ u & \text{otherwise,} \end{cases} \quad (2.11)$$

$$\omega_{F,x \rightarrow a, \ell+1} = \begin{cases} n & \text{if } \omega_{F,b \rightarrow x, \ell} = n \text{ for at least one } b \in \partial x \setminus \{a\}, \\ f & \text{if } \omega_{F,b \rightarrow x, \ell} \neq n \text{ for all } b \in \partial x \setminus \{a\} \text{ and } \omega_{F,b \rightarrow x, \ell} = f \text{ for at least one } b \in \partial x \setminus \{a\}, \\ u & \text{otherwise.} \end{cases} \quad (2.12)$$

In addition to the messages we also define the *mark* of variable node x by letting

$$\omega_{F,x,\ell} = \begin{cases} n & \text{if } \omega_{F,b \rightarrow x, \ell} = n \text{ for at least one } b \in \partial x, \\ f & \text{if } \omega_{F,b \rightarrow x, \ell} \neq n \text{ for all } b \in \partial x \text{ and } \omega_{F,b \rightarrow x, \ell} = f \text{ for at least one } b \in \partial x, \\ u & \text{otherwise.} \end{cases} \quad (2.13)$$

The following statement summarises the relationship between BP and WP.

Fact 2.4. *For all $t \geq 0$ and all x, a we have*

$$\mu_{x \rightarrow a, \ell}(1) = 1/2 \quad \Leftrightarrow \quad \omega_{F,x \rightarrow a, \ell} \neq n, \quad (2.14)$$

$$\mu_{a \rightarrow x, \ell}(1) = 1/2 \quad \Leftrightarrow \quad \omega_{F,a \rightarrow x, \ell} \neq n, \quad (2.15)$$

$$\mu_{x,\ell}(1) = 1/2 \quad \Leftrightarrow \quad \omega_{F,x,\ell} \neq n. \quad (2.16)$$

Moreover, for all $\ell > 2|C(F)|$ we have $\omega_{F,x \rightarrow a, \ell} = \omega_{F,x \rightarrow a, \ell+1}$ and $\omega_{F,a \rightarrow x, \ell} = \omega_{F,a \rightarrow x, \ell+1}$.

Proof. **TOPROVE 2** □

Fact 2.4 implies that the WP messages and marks ‘converge’ in the limit of large ℓ , in the sense that eventually they do not change any more. Let $\omega_{F,x \rightarrow a}, \omega_{F,a \rightarrow x}, \omega_{F,x} \in \{\mathbf{f}, \mathbf{u}, \mathbf{n}\}$ be these limits. Furthermore, let $V_{\mathbf{f}, \ell}(F), V_{\mathbf{u}, \ell}(F), V_{\mathbf{n}, \ell}(F)$ be the sets of variables with the respective mark after $\ell \geq 0$ iterations. Also let $V_{\mathbf{f}}(F), V_{\mathbf{u}}(F), V_{\mathbf{n}}(F)$ be the sets of variables where the limit $\omega_{F,x}$ takes the respective value. The following statement traces WP on the random formula $\mathbf{F}_{\text{DC}, t}$ produced by the decimation process.

Proposition 2.5. *Let $\varepsilon > 0$ and assume that $d > 0, t = t(n) \sim \theta n$ satisfy one of the following conditions:*

- (i) $d < d_{\min}$, or
- (ii) $d > d_{\min}$ and $\theta \notin \{\theta_*, \theta^*\}$.

Then there exists $\ell_0 = \ell_0(d, \theta, \varepsilon) > 0$ such that for any fixed $\ell \geq \ell_0$ with $\lambda = -\log(1 - \theta)$ w.h.p. we have

$$|t + |V_{\mathbf{n}, \ell}(\mathbf{F}_{\text{DC}, t})| - \alpha_* n| < \varepsilon n, \quad |t + |V_{\mathbf{f}, \ell}(\mathbf{F}_{\text{DC}, t})| - (\alpha^* - \alpha_*)n| < \varepsilon n, \quad |V_{\mathbf{n}}(\mathbf{F}_{\text{DC}, t}) \Delta V_{\mathbf{n}, \ell}(\mathbf{F}_{\text{DC}, t})| < \varepsilon n. \quad (2.17)$$

2.4. The check matrix. Since the XOR operation is equivalent to addition modulo two, a XORSAT formula F with variables x_1, \dots, x_n and clauses a_1, \dots, a_m translates into a linear system over \mathbb{F}_2 , as follows. Let A_F be the $m \times n$ -matrix over \mathbb{F}_2 whose (i, j) -entry equals one iff variable x_j appears in clause a_i . Adopting coding parlance, we refer to A_F as the *check matrix* of F . Furthermore, let $y_F \in \mathbb{F}_2^m$ be the vector whose i th entry is one plus the sum of any constant term and the number of negation signs of clause a_i mod two. Then the solutions $\sigma \in \mathbb{F}_2^n$ of the linear system $A_F \sigma = y_F$ are precisely the satisfying assignments of F .

The algebraic properties of A_F therefore have a direct impact on the satisfiability of F . For example, if A_F has rank m , we may conclude immediately that F is satisfiable. Furthermore, the set of solutions of F is an affine subspace of \mathbb{F}_2^n (if non-empty). In effect, if F is satisfiable, then the number of satisfying assignments equals the size of the kernel of A_F . Hence the nullity $\text{nul } A_F = \dim \ker A_F$ of the check matrix is a key quantity.

Indeed, the single most significant ingredient towards turning the heuristic arguments from [25] into rigorous proofs is a formula for the nullity of the check matrix of the XORSAT instance $\mathbf{F}_{\text{DC}, t}$ from the decimation process. To unclutter the notation set $\mathbf{A}_t = A_{\mathbf{F}_{\text{DC}, t}}$. We derive the following proposition from a recent general result about the nullity of random matrices over finite fields [8, Theorem 1.1]. The proposition clarifies the semantics of the function $\Phi_{d, k, \lambda}$ and its maximiser α_{\max} . In physics jargon $\Phi_{d, k, \lambda}$ is known as the Bethe free entropy.

Proposition 2.6. *Let $d > 0$ and $\lambda = -\log(1 - \theta)$. Then*

$$\lim_{n \rightarrow \infty} \text{nul } \mathbf{A}_t = \Phi_{d, k, \lambda}(\alpha_{\max}) \quad \text{in probability.}$$

2.5. Null variables. Proposition 2.6 enables us to derive crucial information about the set of satisfying assignments of $\mathbf{F}_{\text{DC}, t}$. Specifically, for any XORSAT instance F with variables x_1, \dots, x_n let $V_0(F)$ be the set of variables x_i such that $\sigma_i = 0$ for all $\sigma \in \ker A_F$. We call the variables $x_i \in V_0(F)$ *null variables*. Since the set of solutions of F , if non-empty, is a translation of $\ker A_F$, any two solutions σ, σ' of F set the variables in $V_0(F)$ to exactly the same values. The following proposition shows that WP identifies certain variables as null.

Proposition 2.7. *W.h.p. the following two statements are true for any fixed integer $\ell > 0$.*

- (i) *We have $V_{\mathbf{n}, \ell}(\mathbf{F}_{\text{DC}, t}) \subseteq V_0(\mathbf{F}_{\text{DC}, t})$.*
- (ii) *We have $|V_{\mathbf{u}, \ell}(\mathbf{F}_{\text{DC}, t}) \cap V_0(\mathbf{F}_{\text{DC}, t})| = o(n)$.*

Propositions 2.6 and 2.7 enable us to calculate the number of null variables of $\mathbf{F}_{\text{DC}, t}$, so long as we remain clear of the point θ_{cond} where α_{\max} is discontinuous.

Proposition 2.8. *If $\theta \neq \theta_{\text{cond}}$ then $|V_0(\mathbf{F}_{\text{DC}, t})| = \alpha_{\max} n + o(n)$ w.h.p.*

Let us briefly summarise what we have learned thus far. First, because all Belief Propagation messages are half-integral, BP reduces to WP. Second, Proposition 2.5 shows that the fixed points α_*, α^* of $\phi_{d, k, \lambda}$ determine the number of variables marked \mathbf{n} or \mathbf{f} by WP. Third, the function $\Phi_{d, k, \lambda}$ and its maximiser α_{\max} govern the nullity of the check matrix and thereby the number of null variables of $\mathbf{F}_{\text{DC}, t}$. Clearly, the null variables x_i are precisely the ones whose actual marginals $\mathbb{P}[\sigma_{\mathbf{F}_{\text{DC}, t}}(x_i) = s \mid \mathbf{F}_{\text{DC}, t}]$ are *not* uniform. As a next step, we investigate whether BP/WP identify these variables correctly.

In light of Proposition 2.5, in order to investigate the accuracy of BP it suffices to compare the *numbers* of variables marked \mathbf{n} by WP with the true marginals. The following corollary summarises the result.

Corollary 2.9. *For any d, θ the following statements are true.*

- (i) *If $d < d_{\min}$, or $d > d_{\min}$ and $\theta < \theta_{\text{cond}}$, or $d > d_{\min}$ and $\theta > \theta_*$, then $|V_0(\mathbf{F}_{\text{DC},t}) \Delta V_n(\mathbf{F}_{\text{DC},t})| = o(n)$ w.h.p.*
- (ii) *If $d > d_{\min}$ and $\theta_{\text{cond}} < \theta < \theta_*$, then $|V_0(\mathbf{F}_{\text{DC},t}) \Delta V_n(\mathbf{F}_{\text{DC},t})| = \Omega(n)$ w.h.p.*

Thus, so long as $d < d_{\min}$ or $d > d_{\min}$ and $\theta < \theta_{\text{cond}}$ or $\theta > \theta_*$, the BP/WP approximations are mostly correct. By contrast, if $d > d_{\min}$ and $\theta_{\text{cond}} < \theta < \theta_*$, the BP/WP approximations are significantly at variance with the true marginals w.h.p. Specifically, w.h.p. BP deems $\Omega(n)$ frozen variables unfrozen, thereby setting itself up for failure. Indeed, Corollary 2.9 easily implies Theorem 1.3, which in turn implies Theorem 1.1 (ii) without much ado.

In addition, to settle the (non-)reconstruction thresholds set out in Theorem 1.2 we need to investigate the *conditional* marginals given the values of variables at a certain distances from x_{t+1} as in (1.7). This is where the extra value \mathbf{f} from the construction of WP enters. Indeed, for a XORSAT instance F with variables x_1, \dots, x_n and an integer ℓ let $V_{0,\ell}(F)$ be the set of variables x_i such that $\sigma_i = 0$ for all $\sigma \in \ker A_F$ for which $\sigma_h = 0$ for all variables $x_h \in \partial^\ell x_i$. Hence, $V_{0,\ell}(F) \subseteq V_0(F)$ is the set of variables whose ℓ -neighbourhood is contained in $V_0(F)$.

Corollary 2.10. *Assume that $d > d_{\min}$ and let $\varepsilon > 0$.*

- (i) *If $\theta < \theta_{\text{cond}}$, then for any fixed ℓ we have $|V_{\mathbf{f},\ell}(\mathbf{F}_{\text{DC},t}) \cap V_{0,\ell}(\mathbf{F}_{\text{DC},t})| < \varepsilon n$ w.h.p.*
- (ii) *If $\theta > \theta_{\text{cond}}$, then there exists $\ell_0 = \ell_0(d, \theta, \varepsilon)$ such that for any fixed $\ell > \ell_0$ we have*

$$|(V_{n,\ell}(\mathbf{F}_{\text{DC},t}) \cup V_{\mathbf{f},\ell}(\mathbf{F}_{\text{DC},t})) \Delta V_{0,\ell}(\mathbf{F}_{\text{DC},t})| < \varepsilon n \quad \text{w.h.p.}$$

Comparing the number of actually frozen variables with the ones marked \mathbf{f} by WP, we obtain Theorem 1.2.

2.6. Proving BPGD successful. We are left to prove Theorem 1.1. First, we need to compute the (strictly positive) success probability of BPGD for $d < d_{\min}$. At this point, the fact that BPGD has a fair chance of succeeding for $d < d_{\min}$ should not come as a surprise. Indeed, Corollary 2.9 implies that the BP approximations of the marginals are mostly correct for $d < d_{\min}$, at least on the formula $\mathbf{F}_{\text{DC},t}$ created by the decimation process. Furthermore, so long as the marginals are correct, the decimation process $\mathbf{F}_{\text{DC},t}$ and the execution of the BPGD algorithm $\mathbf{F}_{\text{BP},t}$ move in lockstep. The sole difficulty in analysing BPGD lies in proving that the estimates of the algorithm are not just mostly correct, but correct up to only a *bounded* expected number of discrepancies over the entire execution of the algorithm. To prove this fact we combine the method of differential equations with a subtle analysis of the sources of the remaining bounded number of discrepancies. These discrepancies result from the presence of short (i.e., bounded-length) cycles in the graph $G(F)$. Finally, the proof of the second (negative) part of Theorem 1.1 follows by coupling the execution of BPGD with the decimation process, and invoking Theorem 1.3. The details of both arguments can be found in Section 6.

2.7. Discussion. The thrust of the present work is to verify the predictions from [25] on the BPGD algorithm and the decimation process rigorously. Concerning the decimation process, the main gap in the deliberations of Ricci-Tersenghi and Semerjian [25] that we needed to plug is the proof of Proposition 2.8 on the actual number of null variables in the decimation process. The proof of Proposition 2.8, in turn, hinges on the formula for the nullity from Proposition 2.6, whereas Ricci-Tersenghi and Semerjian state the (as it turns out, correct) formulas for the nullity and the number of null variables based on purely heuristic arguments.

Regarding the analysis of the BPGD algorithm, Ricci-Tersenghi and Semerjian state that they rely on the heuristic techniques from the insightful article [11] to predict the formula (1.6), but do not provide any further details; the article [11] principally employs heuristic arguments involving generating functions. By contrast, the method that we use to prove (1.6) is a bit more similar to that of Frieze and Suen [13] for the analysis of a variant of the unit clause algorithm on random k -SAT instances, for which they also obtain the asymptotic success probability. Yet by comparison to the argument of Frieze and Suen, we pursue a more combinatorially explicit approach that demonstrates that certain small sub-formulas that we call ‘toxic cycles’ are responsible for the failure of BPGD. Specifically, the proof of (1.6) combines the method of differential equations with Poissonisation. Finally, the proof of Theorem 1.1 (ii) is an easy afterthought of the analysis of the decimation process.

Yung’s work [27] on random k -XORSAT is motivated by the ‘overlap gap paradigm’ [14], the basic idea behind which is to show that a peculiar clustered geometry of the set of solutions is an obstacle to certain types of algorithms. Specifically, Yung only considers the Unit Clause Propagation algorithm and (a truncated version of) BPGD. Following the path beaten in [20], Yung performs moment computations to establish the overlap gap property. However, moment computations (also called ‘annealed computations’ in physics jargon) only provide one-sided bounds. Yung’s results require spurious lower bounds on the clause length k ($k \geq 9$ for Unit Clause and $k \geq 13$

for BPGD). By contrast, the present proof strategy pivots on the number of null variables rather than overlaps, and Proposition 2.8 provides the precise ‘quenched’ count of null variables. A further improvement over [27] is that the present analysis pinpoints the *precise* threshold up to which BPGD (as well as Unit Clause) succeeds for any $k \geq 3$. Specifically, Yung proves that these algorithms fail for $d > d_{\text{core}}$, while Theorem 1.1 shows that failure occurs already for $d > d_{\text{min}}$ with $d_{\text{min}} < d_{\text{core}}$. Conversely, Theorem 1.1 shows that the algorithms succeed with a non-vanishing probability for $d < d_{\text{min}}$. Thus, Theorem 1.1 identifies the correct threshold for the success of BPGD, as well as the correct combinatorial phenomenon that determines this threshold, namely the onset of reconstruction in the decimation process (Theorems 1.2 and 1.3).

The BPGD algorithm as detailed in Section 2.2 applies to a wide variety of problems beyond random k -XORSAT. Of course, the single most prominent example is random k -SAT. Lacking the symmetries of XORSAT, random k -SAT does not allow for the simplification to discrete messages; in particular, the BP messages are not generally half-integral. In effect, BP and WP are no longer equivalent. In addition to random k -XORSAT, the article [25] also provides a heuristic study of BPGD on random k -SAT. But once again due to the lack of half-integrality, the formulas for the phase transitions no longer come as elegant finite-dimensional expressions. Instead, they now come as infinite-dimensional variational problems. Furthermore, the absence of half-integrality also entails that the present proof strategy does not extend to k -SAT.

The lack of inherent symmetry in random k -SAT can partly be compensated by assuming that the clause length k is sufficiently large (viz. larger than some usually unspecified constant k_0). Under this assumption the random k -SAT version of both the decimation process and the BPGD algorithm have been analysed rigorously [7, 10]. The results are in qualitative agreement with the predictions from [25]. In particular, the BPGD algorithm provably fails to find satisfying assignments on random k -SAT instances even below the threshold where the set of satisfying assignments shatters into well-separated clusters [1, 17]. Furthermore, on random k -SAT a more sophisticated message passing algorithm called Survey Propagation Guided Decimation has been suggested [21, 25]. While on random XORSAT Survey Propagation and Belief Propagation are equivalent, the two algorithms are substantially different on random k -SAT. One might therefore hope that Survey Propagation Guided Decimation outperforms BPGD on random k -SAT and finds satisfying assignments up to the aforementioned shattering transition. A negative result to the effect that Survey Propagation Guided Decimation fails asymptotically beyond the shattering transition point for large enough k exists [15]. Yet a complete analysis of Belief/Survey Propagation Guided Decimation on random k -SAT for any $k \geq 3$ in analogy to the results obtained here for random k -XORSAT remains an outstanding challenge.

Finally, returning to random k -XORSAT, a question for future work may be to investigate the performance of various types of algorithms such as greedy, message passing or local search that aim to find an assignment that violates the least possible number of clauses. Of course, this question is relevant even for $d > d_{\text{sat}}(k)$. A first step based on the heuristic ‘dynamical cavity method’ was recently undertaken by Maier, Behrens and Zdeborová [18].

2.8. Preliminaries and notation. Throughout we assume that $k \geq 3$ and $0 < d < d_{\text{min}}$ and $\theta \in (0, 1)$ are fixed independently of n . We always let $t = t(n) \in \{0, 1, \dots, n\}$ be an integer sequence such that $\lim_{n \rightarrow \infty} t/n = \theta$. Unless specified otherwise we tacitly assume that n is sufficiently large for our various estimates to hold. Asymptotic notation such as $O(\cdot)$ refers to the limit of large n by default, with k, d, θ fixed. We continue to denote by $\alpha_* = \alpha_*(\lambda) = \alpha_*(d, k, \lambda)$ and $\alpha^* = \alpha^*(\lambda) = \alpha^*(d, k, \lambda)$ the smallest/largest fixed points of $\phi_{d,k,\lambda}$ in $[0, 1]$ and by $\lambda_* = \lambda_*(d, k)$, $\lambda^* = \lambda^*(d, k)$, $\theta_* = \theta_*(d, k)$, $\theta^* = \theta^*(d, k)$ the quantities defined in (1.9)–(1.10).

For a formula F and a partial assignment $\sigma : U \rightarrow \{0, 1\}$ with $U \subseteq V(F)$ let $F[\sigma]$ be the simplified formula obtained by substituting constants for the variables in U . The *length* of a clause of $F[\sigma]$ is defined as the number of variables from $V(F) \setminus U$ that the clause contains.

The following fact provides the correctness of BP on formulas represented by acyclic graphs $G(F)$.

Fact 2.11 ([19, Chapter 14]). *For a XORSAT Formula F with an acyclic bipartite graph $G(F)$ the BP marginals as defined in (2.7) are exact, i.e.*

$$\lim_{\ell \rightarrow \infty} \mu_{F,x,\ell}(1) = \mathbb{P}[\sigma_F(x) = 1].$$

2.9. Organisation. The rest of the paper is organised as follows. Section 3 contains the proof of Proposition 2.2. Subsequently in Section 4 we investigate Warning Propagation to prove Propositions 2.5 and 2.7. Furthermore, Section 5 deals with the study of the check matrix; here we prove Propositions 2.6 and 2.8 as well as Corollaries 2.9

and 2.10. Additionally, with all these preparations completed we put all the pieces together to complete the proofs of Theorems 1.2 and 1.3 in Section 5.5. Finally, Section 6 contains the proof of Theorem 1.1.

3. PROOF OF PROPOSITION 2.2

Even though a few steps are mildly intricate, the proof of Proposition 2.2 mostly consists of ‘routine calculus’. As a convenient shorthand we introduce

$$\zeta_\lambda(z) = \zeta_{d,k,\lambda}(z) = \phi_{d,k,\lambda}(z) - z = 1 - \exp(-\lambda - dz^{k-1}) - z.$$

Its derivatives read

$$\zeta'_\lambda(z) = d(k-1) z^{k-2} \exp(-\lambda - dz^{k-1}) - 1 \quad \text{and} \quad (3.1)$$

$$\zeta''_\lambda(z) = d(k-1) z^{k-3} \exp(-\lambda - dz^{k-1}) \left[(k-2) - d(k-1) z^{k-1} \right]. \quad (3.2)$$

Also let

$$z_0 = z_0(d, k) = \left(\frac{k-2}{d(k-1)} \right)^{\frac{1}{k-1}}. \quad (3.3)$$

We begin by investigating the zeros of ζ_λ , obviously identical with fixed points of $\phi_{d,k,\lambda}$.

Lemma 3.1. *Assume that $\lambda > 0$.*

- (i) *The function ζ_λ has either one or three zeros in $z \in [0, 1]$, possibly including multiple zeros. If ζ_λ has three zeros, then at least one lies in the interval $[0, z_0]$ and at least one lies in the interval $[z_0, 1]$.*
- (ii) *Also, ζ_λ has at most two stationary points, a minimum and a maximum, and if it has both, the minimum occurs left of the maximum.*
- (iii) *If ζ_λ has a unique zero, then α_* is a stable fixed point of $\phi_{d,k,\lambda}$ and $\sup_{z \in [0,1]} \phi'_{d,k,\lambda}(z) < 1$.*
- (iv) *If ζ_λ has three zeros but no double zero, then α_*, α^* are stable fixed points of $\phi_{d,k,\lambda}$. Additionally, $\phi_{d,k,\lambda}$ possesses an unstable fixed point $\alpha_u \in (\alpha_*, \alpha^*)$. Furthermore, there exists $\varepsilon = \varepsilon(d, \lambda) > 0$ such that*

$$\sup_{z \in [0, \alpha_* + \varepsilon]} \phi'_{d,k,\lambda}(z) < 1, \quad \sup_{z \in [\alpha^* - \varepsilon, 1]} \phi'_{d,k,\lambda}(z) < 1.$$

Proof. **TOPROVE 3** □

The following statement implies that $\phi_{d,k,\lambda}$ has only a single fixed point if $d < d_{\min}$.

Lemma 3.2. *Let $\lambda > 0$. If $d < d_{\min}$, then ζ_λ has a unique zero and is strictly decreasing.*

Proof. **TOPROVE 4** □

Proceeding to average degrees $d > d_{\min}$, we verify that the values λ_*, λ^* from Section 1.5 are well defined and satisfy the inequality (1.9).

Lemma 3.3. *If $d > d_{\min}$, then the polynomial $d(k-1)z^{k-2}(1-z) - 1$ has precisely two roots $0 < z_* < z^* < 1$ and the values λ_*, λ^* defined in (1.9) satisfy $\lambda_* > \lambda^*$. Furthermore, $d_{\text{core}} > d_{\min}$ and $\lambda^* = 0$ iff $d \geq d_{\text{core}}$.*

Proof. **TOPROVE 5** □

We are ready to identify the zeros of ζ_λ for $d > d_{\min}$, depending on the regime of λ .

Lemma 3.4. *Let $\lambda > 0$ and assume that $d > d_{\min}$.*

- (i) *If $\lambda < \lambda^*$, then ζ_λ has a unique zero.*
- (ii) *If $\lambda^* < \lambda < \lambda_*$, then ζ_λ has three distinct zeros.*
- (iii) *If $\lambda > \lambda_*$, then ζ_λ has a unique zero.*

Proof. **TOPROVE 6** □

Combining Lemmas 3.3 and 3.4, we can now verify the analytic properties of the function $\lambda \mapsto \alpha_*$ and $\lambda \mapsto \alpha^*$.

Lemma 3.5. *Let $0 < d < d_{\text{sat}}$ and $\lambda > 0$.*

(i) If $d < d_{\min}$, then the function $\lambda \in (0, \infty) \mapsto \alpha_* = \alpha^*$ is analytic with derivative

$$\frac{\partial \alpha_*}{\partial \lambda} = \frac{1 - \alpha_*}{1 - d(k-1)\alpha_*^{k-2}(1 - \alpha_*)} < 1. \quad (3.4)$$

(ii) If $d > d_{\min}$, then $\lambda \in (0, \lambda_*) \mapsto \alpha_*$ is analytic with derivative (3.4).

(iii) If $d > d_{\min}$, then $\lambda \in (\lambda^*, \infty) \mapsto \alpha^*$ is analytic differentiable with derivative

$$\frac{\partial \alpha^*}{\partial \lambda} = \frac{1 - \alpha^*}{1 - d(k-1)\alpha^{*k-2}(1 - \alpha^*)}.$$

Proof. **TOPROVE 7** □

As a final preparation towards the proof of Proposition 2.2 we investigate the solution λ_{cond} to the differential equation (1.11); notice that Lemma 3.5 shows that this ODE does indeed possess a unique solution on $(d_{\min}, d_{\text{sat}}]$.

Lemma 3.6. *For any $0 < d < d_{\text{sat}}$ we have $0 < \lambda_{\text{cond}} < \lambda_*$. Furthermore, for all $0 < \lambda < \lambda_{\text{cond}}$ we have $\Phi_{d,k,\lambda}(\alpha_*) > \Phi_{d,k,\lambda}(\alpha^*)$, while $\Phi_{d,k,\lambda}(\alpha_*) < \Phi_{d,k,\lambda}(\alpha^*)$ for $\lambda_* > \lambda > \lambda_{\text{cond}}$.*

Proof. **TOPROVE 8** □

Proof. **TOPROVE 9** □

4. WARNING PROPAGATION AND LOCAL WEAK CONVERGENCE

In this section we prove Propositions 2.5 and 2.7. The proofs rely on the concept of local weak convergence. Specifically, we are going to set up a Galton-Watson process that mimics the local topology of the graph $G(\mathbf{F}_{\text{DC},t})$ up to any fixed depth ℓ . Subsequently we will analyse WP on the Galton-Watson tree and argue that the result extends to $G(\mathbf{F}_{\text{DC},t})$.

4.1. Local weak convergence. The construction of the Galton-Watson process $\mathbb{T} = \mathbb{T}(d, k, t)$ is pretty straightforward. The process has two types called *variable nodes* and *check nodes*. The process starts with a single variable node v_0 . Furthermore, each variable node begets a $\text{Po}(d)$ number of check nodes as offspring, while the offspring of a check node is a $\text{Bin}(k-1, 1-t/n)$ number of variable nodes.

Let \mathbb{T} be the Galton-Watson tree rooted at v_0 that this process generates; \mathbb{T} may be infinite. Hence, for an integer ℓ obtain $\mathbb{T}^{(\ell)}$ from \mathbb{T} by deleting all variable/check nodes at distance greater than 2ℓ from v_0 . Thus, $\mathbb{T}^{(\ell)}$ is a finite random tree rooted at v_0 . For any graphs T, T' rooted at v, v' , respectively, we write $T \cong T'$ if there is a graph isomorphism $\iota: T \rightarrow T'$ such that $\iota(v) = v'$. Furthermore, for a vertex v of $G(\mathbf{F}_{\text{DC},t})$ and an integer ℓ we let $\partial_{\mathbf{F}_{\text{DC},t}}^{\leq \ell} v$ be the subgraph obtained from $G(\mathbf{F}_{\text{DC},t})$ by deleting all vertices at distance greater than 2ℓ from v , rooted at v . Finally, for a rooted graph g and an integer ℓ we let $N_t^{(\ell)}(g)$ be the number of vertices v of $G(\mathbf{F}_{\text{DC},t})$ such that $\partial_{\mathbf{F}_{\text{DC},t}}^{\leq \ell} v \cong g$.

Lemma 4.1. *For any rooted tree g we have*

$$\mathbb{E} \left| N_t^{(\ell)}(g) - (n-t) \mathbb{P} \left[\mathbb{T}^{(\ell)} \cong g \right] \right| = o(n). \quad (4.1)$$

Proof. **TOPROVE 10** □

4.2. Proof of Proposition 2.5. To prove Proposition 2.5 we estimate the sizes $|V_{\mathbf{n},\ell}(\mathbf{F}_{\text{DC},t})|, |V_{\mathbf{f},\ell}(\mathbf{F}_{\text{DC},t})|$ separately. Recall that $\theta \sim t/n$.

Lemma 4.2. *Let $\varepsilon > 0$ and assume that one of the following conditions is satisfied:*

- (i) $d < d_{\min}$, or
- (ii) $d > d_{\min}$ and $|\theta_* - \theta| > \varepsilon$.

Then there exists $\ell_0 = \ell_0(d, \varepsilon) > 0$ such that for any fixed $\ell \geq \ell_0$ with $\lambda = -\log(1 - \theta)$ w.h.p. we have

$$|t + |V_{\mathbf{n},\ell}(\mathbf{F}_{\text{DC},t})| - \alpha_* n| < \varepsilon n.$$

Proof. **TOPROVE 11** □

Lemma 4.3. *Let $\varepsilon > 0$ and assume that $d > 0$, $t = t(n)$ are such that one of the following conditions is satisfied:*

- (i) $d < d_{\min}$, or

(ii) $d > d_{\min}$, $|\theta_* - \theta| > \varepsilon$ and $|\theta^* - \theta| > \varepsilon$.

Then there exists $\ell_0 = \ell_0(d, \varepsilon) > 0$ such that for any fixed $\ell \geq \ell_0$ with $\lambda = -\log(1 - \theta)$ w.h.p. we have

$$|V_{\mathbf{f}, \ell}(\mathbf{F}_{\text{DC}, t}) - (\alpha^* - \alpha_*)n| < \varepsilon n.$$

Proof. **TOPROVE 12** □

Finally, we compare the set $V_{\mathbf{n}, \ell}(\mathbf{F}_{\text{DC}, t})$ obtained after a (large but) bounded number of iterations with the ultimate sets $V_{\mathbf{n}}(\mathbf{F}_{\text{DC}, t})$ obtained upon convergence of WP. The proof of the following lemma is an adaptation of the argument from [23] for cores of random hypergraphs.

Lemma 4.4. *Assume that $\theta \in (0, 1) \setminus \{\theta_*, \theta^*\}$. Then for any $\varepsilon > 0$ there exists $\ell_0 = \ell_0(d, \theta, \varepsilon)$ such that for all $\ell > \ell_0$ we have $|V_{\mathbf{n}, \ell}(\mathbf{F}_{\text{DC}, t}) \Delta V_{\mathbf{n}}(\mathbf{F}_{\text{DC}, t})| < \varepsilon n$ w.h.p.*

Proof. **TOPROVE 13** □

Proof. **TOPROVE 14** □

4.3. Proof of Proposition 2.7. We deal with the two claims separately. Towards the first claim we establish the following stronger, deterministic statement.

Lemma 4.5. *For any XORSAT instance F with variables $V_n = \{x_1, \dots, x_n\}$ and any integer $\ell \geq 0$ we have $V_{\mathbf{n}, \ell}(F) \subseteq V_0(F)$.*

Proof. **TOPROVE 15** □

The following lemma deals with the variables that WP marks u.

Lemma 4.6. *For any fixed $\ell \geq 0$ we have $|V_{\mathbf{u}, \ell}(\mathbf{F}_{\text{DC}, t}) \cap V_0(\mathbf{F}_{\text{DC}, t})| = o(n)$ w.h.p.*

Proof. **TOPROVE 16** □

Proof. **TOPROVE 17** □

5. ANALYSIS OF THE CHECK MATRIX

In this section we prove Propositions 2.6 and 2.8. Proposition 2.6 is an easy consequence of [8, Theorem 1.1]. Furthermore, Proposition 2.8 follows from Proposition 2.6 by interpolating on the parameter λ ; a related argument was recently used in [9] to show that certain random combinatorial matrices have full rank w.h.p. In addition, we prove Corollaries 2.9 and 2.10 and subsequently complete the proofs of Theorems 1.2–1.3.

5.1. Proof of Proposition 2.6. We use a general result [8, Theorem 1.1] about the rank of sparse random matrices from a fairly universal class of distributions. The definition of this general random matrix goes as follows. Let $\mathfrak{d}, \mathfrak{k} \geq 0$ be integer-values random variables such that $0 < \mathbb{E}[\mathfrak{d}^3] + \mathbb{E}[\mathfrak{k}^3] < \infty$. Moreover, let $(\mathfrak{d}_i, \mathfrak{k}_i)_{i \geq 0}$ be families of mutually independent random variables such that $\mathfrak{d}_i \stackrel{\text{dist}}{=} \mathfrak{d}$ and $\mathfrak{k}_i \stackrel{\text{dist}}{=} \mathfrak{k}$. Let $\bar{\mathfrak{d}} = \mathbb{E}[\mathfrak{d}]$ and $\bar{\mathfrak{k}} = \mathbb{E}[\mathfrak{k}]$ and for an integer $n > 0$ let $\mathfrak{m} = \mathfrak{m}_n \stackrel{\text{dist}}{=} \text{Po}(\bar{\mathfrak{d}}n/\bar{\mathfrak{k}})$. The sequence $(\mathfrak{m}_n)_n$ is independent of $(\mathfrak{d}_i, \mathfrak{k}_i)_{i \geq 0}$. Further, let \mathfrak{S}_n be the event that

$$\sum_{i=1}^n \mathfrak{d}_i = \sum_{i=1}^{\mathfrak{m}_n} \mathfrak{k}_i. \quad (5.1)$$

It is a known fact that $\mathbb{P}[\mathfrak{S}_n] = \Omega(n^{-1/2})$ [8, Proposition 1.10]. Given that \mathfrak{S}_n occurs, create a simple random bipartite graph \mathfrak{G}_n with a set $\mathfrak{V}_n = \{\mathfrak{x}_1, \dots, \mathfrak{x}_n\}$ of *variable nodes* and a set $\mathfrak{C}_n = \{\mathfrak{c}_1, \dots, \mathfrak{c}_{\mathfrak{m}_n}\}$ of *check nodes* uniformly at random subject to the condition that \mathfrak{x}_j has degree \mathfrak{d}_j and \mathfrak{c}_i has degree \mathfrak{k}_i for all $1 \leq j \leq n$ and $1 \leq i \leq \mathfrak{m}_n$. Finally, let \mathfrak{A}_n be the biadjacency matrix of \mathfrak{G}_n . Thus, \mathfrak{A}_n has size $\mathfrak{m}_n \times n$ and its (i, j) -entry equals 1 iff \mathfrak{x}_j and \mathfrak{c}_i are adjacent in \mathfrak{G}_n .

Theorem 5.1 (special case of [8, Theorem 1.1]). *Let $\mathfrak{D}(z) = \sum_{h=0}^{\infty} \mathbb{P}[\mathfrak{d} = h] z^h$ and $\mathfrak{K}(z) = \sum_{h=0}^{\infty} \mathbb{P}[\mathfrak{k} = h] z^h$ be the probability generating functions of $\mathfrak{d}, \mathfrak{k}$, respectively. Furthermore, let*

$$\mathfrak{F}: [0, 1] \rightarrow \mathbb{R}, \quad z \mapsto \mathfrak{D}(1 - \mathfrak{K}'(z)/\mathfrak{K}'(1)) - \frac{\mathfrak{D}'(1)}{\mathfrak{K}'(1)}(1 - \mathfrak{K}(z) - (1 - z)\mathfrak{K}'(z)). \quad (5.2)$$

Then

$$\lim_{n \rightarrow \infty} \frac{1}{n} \text{nul } \mathfrak{A}_n = \max_{z \in [0, 1]} \mathfrak{F}(z) \quad \text{in probability.}$$

We now derive Proposition 2.6 from Theorem 5.1 by identifying suitable distributions $\mathfrak{D}, \mathfrak{E}$ such that \mathfrak{A}_n resembles \mathbf{A}_t .

Proof. **TOPROVE 18** □

5.2. Proof of Proposition 2.8. We continue to work with the random matrix $\mathbf{A}[\lambda]$ from the above proof of Proposition 2.6. As we recall, this matrix is obtained by adding $\mathbf{l} = \text{Po}(\lambda n)$ stochastically independent new rows to the matrix $\mathbf{A}(\mathbf{F})$ that each contain a single non-zero entry in a uniformly random position. Combining (??)–(??), we see that

$$|\mathbb{E}[\text{nul } \mathbf{A}[\lambda]] - \mathbb{E}[\text{nul } \mathbf{A}_t]| = o(n) \quad \text{for } \lambda = -\log(1 - \theta). \quad (5.3)$$

Towards the proof of Proposition 2.8 we observe that $\text{nul } \mathbf{A}[\lambda], \text{nul } \mathbf{A}_t$ concentrate about their expectations.

Lemma 5.2. *We have*

$$\mathbb{P} [|\text{nul } \mathbf{A}_t - \mathbb{E}[\text{nul } \mathbf{A}_t]| > \sqrt{n} \log n] = o(n^{-10}), \quad \mathbb{P} [|\text{nul } \mathbf{A}[\lambda] - \mathbb{E}[\text{nul } \mathbf{A}[\lambda]]| > \sqrt{n} \log n] = o(n^{-10}). \quad (5.4)$$

Proof. **TOPROVE 19** □

We are going to estimate $|V_0(\mathbf{F}_{\text{DC},t})|$ by way of estimating changes of $\text{nul } \mathbf{A}[\lambda]$ as λ varies. Since $\text{nul } \mathbf{A}[\lambda]/n$ converges to $\Phi_{d,k,\lambda}(\alpha_{\max})$ by Proposition 2.6, we thus need to estimate the derivative $\frac{\partial}{\partial \lambda} \Phi_{d,k,\lambda}(\alpha_{\max})$.

Lemma 5.3. *Let $d > 0$ and assume that*

- (i) $d < d_{\min}$, or
- (ii) $d > d_{\min}$ and $\lambda \in (0, \infty) \setminus \{\lambda_{\text{cond}}\}$.

Then

$$\frac{\partial}{\partial \lambda} \Phi_{d,k,\lambda}(\alpha_{\max}) = \alpha_{\max} - 1. \quad (5.5)$$

Proof. **TOPROVE 20** □

Complementing the analytic formula (5.5), we now derive a combinatorial interpretation of the derivative of the nullity. For a matrix \mathbf{A} of size $m \times n$ let $V_0(\mathbf{A})$ be the set of all indices $i \in [n]$ such that $\sigma_i = 0$ for all $\sigma \in \ker \mathbf{A}$.

Lemma 5.4. *For any $d, \lambda > 0$ we have*

$$\frac{\partial}{\partial \lambda} \mathbb{E}[\text{nul } \mathbf{A}[\lambda]] = \frac{\mathbb{E}[|V_0(\mathbf{A}[\lambda])|]}{n} - 1.$$

Proof. **TOPROVE 21** □

With these preparations in place we can now derive the desired formulas for $|V_0(\mathbf{A}_t)|$. We treat the cases $\alpha_{\max} = \alpha_*$ and $\alpha_{\max} = \alpha^*$ separately.

Lemma 5.5. *Assume that $d, \lambda > 0$ satisfy*

$$\Phi_{d,k,\lambda}(\alpha_*) > \Phi_{d,k,\lambda}(\alpha) \quad \text{for all } \alpha \in [0, 1] \setminus \{\alpha_*\}. \quad (5.6)$$

Then $|V_0(\mathbf{A}[\lambda])| = \alpha_ n + o(n)$ w.h.p.*

Proof. **TOPROVE 22** □

Lemma 5.6. *Assume that $d, \lambda > 0$ are such that*

$$\Phi_{d,k,\lambda}(\alpha^*) > \Phi_{d,k,\lambda}(\alpha) \quad \text{for all } \alpha \in [0, 1] \setminus \{\alpha^*\}. \quad (5.7)$$

Then $|V_0(\mathbf{A}[\lambda])| = \alpha^ n + o(n)$ w.h.p.*

Proof. **TOPROVE 23** □

Proof. **TOPROVE 24** □

5.3. **Proof of Corollary 2.9.** There are four cases to consider separately. Let $\varepsilon > 0$.

Case 1: $d < d_{\min}$: As Proposition 2.2 (i) shows, in this case we have $\alpha_* = \alpha^*$ for all $\lambda > 0$; thus, the function $\phi_{d,k,\lambda}$ has only the single fixed point α_* , which is stable. Furthermore, Proposition 2.5 shows that $||V_{n,\ell}(\mathbf{F}_{\text{DC},t})| - \alpha_* n| < \varepsilon n/2$ for large enough ℓ w.h.p. Moreover, Proposition 2.8 yields $|V_0(\mathbf{F}_{\text{DC},t})| = \alpha_* n + o(n)$ w.h.p. Therefore, Proposition 2.7 implies that $|V_0(\mathbf{F}_{\text{DC},t}) \Delta V_{n,\ell}(\mathbf{F}_{\text{DC},t})| < \varepsilon n$ w.h.p. for large enough ℓ . Since $|V_{n,\ell}(\mathbf{F}_{\text{DC},t})| \subseteq V_0(\mathbf{F}_{\text{DC},t})$ w.h.p. and $|V_{n,\ell}(\mathbf{F}_{\text{DC},t}) \Delta V_n(\mathbf{F}_{\text{DC},t})| < \varepsilon n$ by (2.17), the assertion follows.

Case 2: $d_{\min} < d < d_{\text{sat}}$ and $\theta > \theta_*$: A similar argument as under Case 1 applies. Indeed, Proposition 2.2 (ii) shows that $\alpha_* = \alpha^*$ is the unique and stable fixed point of $\phi_{d,k,\lambda}$. Since $||V_{n,\ell}(\mathbf{F}_{\text{DC},t})| - \alpha_* n| < \varepsilon n/2$ for large ℓ w.h.p. by Proposition 2.5 and $|V_0(\mathbf{F}_{\text{DC},t})| = \alpha_* n + o(n)$ w.h.p. by Proposition 2.8, Proposition 2.7 yields $|V_0(\mathbf{F}_{\text{DC},t}) \Delta V_{n,\ell}(\mathbf{F}_{\text{DC},t})| < \varepsilon n$ w.h.p. Therefore, (2.17) implies the assertion.

Case 3: $d_{\min} < d < d_{\text{sat}}$ and $\theta < \theta_{\text{cond}}$: Proposition 2.2 (ii) shows that $\alpha_* < \alpha^*$ in this case. Moreover, Proposition 2.5 yields $||V_{n,\ell}(\mathbf{F}_{\text{DC},t})| - \alpha_* n| < \varepsilon n/2$ for large ℓ w.h.p., while Proposition 2.8 and Proposition 2.2 (iii) imply that $|V_0(\mathbf{F}_{\text{DC},t})| = \alpha_* n + o(n)$ w.h.p. Thus, the same steps as in Cases 1–2 complete the proof.

Case 4: $d_{\min} < d < d_{\text{sat}}$ and $\theta_{\text{cond}} < \theta < \theta_*$: Once again Proposition 2.2 (ii) shows that $\alpha_* < \alpha^*$, Proposition 2.5 yields $||V_{n,\ell}(\mathbf{F}_{\text{DC},t})| - \alpha_* n| < \varepsilon n/2$ for large ℓ w.h.p., and Proposition 2.8 and Proposition 2.2 (iii) show that $|V_0(\mathbf{F}_{\text{DC},t})| = \alpha^* n + o(n)$ w.h.p. Since $|V_{n,\ell}(\mathbf{F}_{\text{DC},t})| \subseteq V_0(\mathbf{F}_{\text{DC},t})$ w.h.p., the assertion follows from (2.17) and the fact that $\alpha_* < \alpha^*$.

5.4. **Proof of Corollary 2.10.** Assume first that $\theta < \theta_{\text{cond}}$. Then Corollary 2.9 shows that $|V_0(\mathbf{F}_{\text{DC},t}) \Delta V_n(\mathbf{F}_{\text{DC},t})| = o(n)$ for large enough ℓ . Since $V_n(\mathbf{F}_{\text{DC},t}) \cap V_f(\mathbf{F}_{\text{DC},t}) = \emptyset$ by construction, the first assertion follows.

Now suppose $\theta > \theta_{\text{cond}}$. Then Proposition 2.5 yields $||V_{f,\ell}(\mathbf{F}_{\text{DC},t})| - \alpha^* n| < \varepsilon n/2$ for large ℓ w.h.p., while Proposition 2.8 and Proposition 2.2 (iii) show that $|V_0(\mathbf{F}_{\text{DC},t})| = \alpha^* n + o(n)$ w.h.p. Additionally, Proposition 2.5 shows that $|V_{u,\ell}(\mathbf{F}_{\text{DC},t}) \cap V_0(\mathbf{F}_{\text{DC},t})| < \varepsilon n$ for large ℓ , which implies the assertion.

5.5. **Proofs of Theorems 1.2 and 1.3.** We begin with the following observation.

Lemma 5.7. *Let $\sigma \in \ker(\mathbf{F}_{\text{DC},t})$ be uniformly random. For any $\ell > 0$ w.h.p. we have*

$$\mathbb{P} \left[\sigma_{x_{t+1}} = 0 \mid \mathbf{F}_{\text{DC},t}, \sigma_{\partial^{2\ell} x_{t+1}} \right] = \frac{1}{2} \left(1 + \mathbb{1}\{x_{t+1} \in V_{f,\ell}(\mathbf{F}_{\text{DC},t}) \cup V_{n,\ell}(\mathbf{F}_{\text{DC},t})\} \right), \quad (5.8)$$

$$\pi_{\mathbf{F}_{\text{DC},t}} = \mathbb{P} \left[\sigma_{x_{t+1}} = 0 \mid \mathbf{F}_{\text{DC},t} \right] = \frac{1}{2} \left(1 + \mathbb{1}\{x_{t+1} \in V_0(\mathbf{F}_{\text{DC},t})\} \right). \quad (5.9)$$

Proof. **TOPROVE 25** □

Proof. **TOPROVE 26** □

Proof. **TOPROVE 27** □

6. BELIEF PROPAGATION GUIDED DECIMATION

In this section we prove Theorem 1.1. We begin by arguing that BPGD is actually equivalent to the simple combinatorial Unit Clause Propagation algorithm. Then we prove the ‘positive’ part, i.e., the formula (1.6) for the success probability for $d < d_{\min}$. Subsequently we prove the second part of the theorem concerning $d_{\min} < d < d_{\text{sat}}$.

6.1. Unit Clause Propagation redux. The simple-minded Unit Clause Propagation algorithm attempts to assign random values to as yet unassigned variables one after the other. After each such random assignment the algorithm pursues the ‘obvious’ implications of its decisions. Specifically, the algorithm substitutes its chosen truth values for all occurrences of the already assigned variables. If this leaves a clause with only a single unassigned variable, a so-called ‘unit clause’, the algorithm assigns that variable so as to satisfy the unit clause. If a conflict occurs because two unit clauses impose opposing values on a variable, the algorithm declares that a conflict has occurred, sets the variable to false and continues; of course, in the event of a conflict the algorithm will ultimately fail to produce a satisfying assignment. The pseudocode for the algorithm is displayed in Algorithm 3.

Let $F_{\text{UC},t}$ denote the simplified formula obtained after the first t iterations (in which the truth values chosen for x_1, \dots, x_t and any values implied by Unit Clauses have been substituted). We notice that the values assigned during Steps 6–12 are deterministic consequences of the choices in Step 5. In particular, the order in which unit clauses are processed Steps 6–12 does not affect the output of the algorithm.


```

1 Let  $U = \emptyset$  and let  $\sigma_{UC} : U \rightarrow \{0, 1\}$  be the empty assignment;
2 for  $t = 0, \dots, n - 1$  do
3   if  $x_{t+1} \notin U$  then
4     add  $x_{t+1}$  to  $U$ ;
5     choose  $\sigma_{UC}(x_{t+1}) \in \{0, 1\}$  uniformly at random;
6     while  $F[\sigma_{UC}]$  contains a unit clause  $a$  do
7       let  $x$  be the variable in  $a$ ;
8       let  $s \in \{0, 1\}$  be the truth value that  $x$  needs to take to satisfy  $a$ ;
9       if another unit clause  $a'$  exists that requires  $x$  be set to  $1 - s$  then
10        output 'conflict' and let  $\sigma_{UC}(x) = 0$ ;
11      else
12        add  $x$  to  $U$  and let  $\sigma_{UC}(x) = s$ ;
13 return  $\sigma_{UC}$ ;

```

Algorithm 3: The UCP algorithm.

Proposition 6.1. *We have*

$$\mathbb{P}[\text{BPGD outputs a satisfying assignment of } F] = \mathbb{P}[\text{UCP outputs a satisfying assignment of } F].$$

Proof. **TOPROVE 28** □

In light of Proposition 6.1 we are left to study the success probability of UCP. The following two subsections deal with this task for $d < d_{\min}$ and $d > d_{\min}$, respectively.

6.2. The success probability of UCP for $d < d_{\min}$. We continue to denote by $F_{UC,t}$ the sub-formula obtained after the first t iterations of UCP. Let $V_n = \{x_1, \dots, x_n\}$ be the set of variables of the XORSAT instance F . Also, let $V(t) \subseteq \{x_{t+1}, \dots, x_n\}$ be the set of variables of $F_{UC,t}$. Thus, $V(t)$ contains those variables among x_{t+1}, \dots, x_n whose values are not implied by the assignment of x_1, \dots, x_t via unit clauses. Also let $C(t)$ be the set of clauses of $F_{UC,t}$; these clauses contain variables from $V(t)$ only, and each clause contains at least two variables. Let $\bar{V}(t) = V_n \setminus V(t)$ be the set of assigned variables. Thus, after its first t iterations UCP has constructed an assignment $\sigma_{UC} : \bar{V}(t) \rightarrow \{0, 1\}$. Moreover, let $V'(t+1) = V(t) \setminus V(t+1)$ be the set of variables that receive values in the course of the iteration $t+1$ for $0 \leq t < n$. Additionally, let $C'(t+1)$ be the set of clauses of $F_{UC,t}$ that consists of variables from $V'(t+1)$ only. Finally, let $F'_{UC,t+1}$ be the formula comprising the variables $V'(t+1)$ and the clauses $C'(t+1)$.

To characterise the distribution of $F_{UC,t}$ let $\mathbf{n}(t) = |V(t)|$ and let $\mathbf{m}_\ell(t)$ be the number of clauses of length ℓ , i.e., clauses that contain precisely ℓ variables from $V(t)$. Observe that $\mathbf{m}_1(t) = 0$ because unit clauses get eliminated. Let \mathfrak{F}_t be the σ -algebra generated by $\mathbf{n}(t)$ and $(\mathbf{m}_\ell(t))_{2 \leq \ell \leq k}$.

Fact 6.2. *The XORSAT formula $F_{UC,t}$ is uniformly random given \mathfrak{F}_t . In other words, the variables that appear in each clause are uniformly random and independent, as are their signs.*

Proof. **TOPROVE 29** □

We proceed to estimate the random variables $\mathbf{n}(t), \mathbf{m}_\ell(t)$. Let $\alpha(t) = |\bar{V}(t)|/n$ so that $\mathbf{n}(t) = n(1 - \alpha(t))$. Recall, that $\bar{V}(t) = V_n \setminus V(t)$. Let $\lambda = \lambda(\theta) = -\log(1 - \theta)$ with $\theta \sim t/n$ and recall that $\alpha_* = \alpha_*(d, k, \lambda)$ denotes the smallest fixed point of $\phi_{d,k,\lambda}$. The proof of the following proposition proof can be found in Section 6.2.1.

Proposition 6.3. *Suppose that $d < d_{\min}(k)$. There exists a function $\delta = \delta(n) = o(1)$ such that for all $0 \leq t < n$ and all $2 \leq \ell \leq k$ we have*

$$\mathbb{P}[|\alpha(t) - \alpha_*| > \delta] = O(n^{-2}), \quad \mathbb{P}\left[\left|\mathbf{m}_\ell(t) - \frac{dn}{k} \binom{k}{\ell} (1 - \alpha_*)^\ell \alpha_*^{k-\ell}\right| > \delta n\right] = O(n^{-2}). \quad (6.1)$$

Proposition 6.3 paves the way for the actual computation of the success probability of UCP. Let \mathcal{R}_t be the event that a conflict occurs in iteration t . The following proposition gives us the correct value of $\mathbb{P}[\mathcal{R}_t | \mathfrak{F}_t]$ w.h.p. Since \mathfrak{F}_t is a random variable the value for the probability $\mathbb{P}[\mathcal{R}_t | \mathfrak{F}_t]$ is random as well.

Proposition 6.4. Fix $\varepsilon > 0$, let $0 \leq t < (1 - \varepsilon)n$ and define

$$f_n(t) = d(k-1)(1 - \alpha_*)\alpha_*^{k-2}. \quad (6.2)$$

Then with probability $1 - o(1/n)$ we have

$$\mathbb{P}[\mathcal{R}_t \mid \mathfrak{F}_t] = \frac{f_n(t)^2}{4(n-t)(1-f_n(t))^2} + o(1/n).$$

The proof of Proposition 6.4 can be found in Section 6.2.2. Moreover, in Section 6.2.3 we prove the following.

Proposition 6.5. Fix $\varepsilon > 0$ and $\ell \geq 1$. For any $0 \leq t_1 < \dots < t_\ell < (1 - \varepsilon)n$ we have

$$\mathbb{P}\left[\bigcap_{i=1}^{\ell} \mathcal{R}_{t_i}\right] \sim \prod_{i=1}^{\ell} \frac{f_n(t_i)^2}{4(n-t_i)(1-f_n(t_i))^2}. \quad (6.3)$$

Finally, the following statement deals with the εn final steps of the algorithm.

Proposition 6.6. For any $\delta > 0$ there exists $\varepsilon > 0$ such that $\mathbb{P}[\bigcup_{(1-\varepsilon)n < t < n} \mathcal{R}_t] < \delta$.

Before we proceed we notice that Propositions 6.4–6.6 imply the first part of Theorem 1.1.

Proof. **TOPROVE 30** □

6.2.1. *Proof of Proposition 6.3.* The proof of Proposition 6.3 is based on the method of differential equations. Specifically, based on Fact 6.2 we derive a system of ODEs that track the random variables $\alpha(t), \mathbf{m}_2(t), \dots, \mathbf{m}_k(t)$. We will then identify the unique solution to this system. As a first step we work out the conditional expectations of $\alpha(t+1), \mathbf{m}_2(t+1), \dots, \mathbf{m}_k(t+1)$ given \mathfrak{F}_t .

Lemma 6.7. If $2\mathbf{m}_2(t)/\mathbf{n}(t) < 1 - \Omega(1)$ and $\mathbf{n}(t) = \Omega(n)$, then

$$\mathbb{E}[\mathbf{n}(t+1) - \mathbf{n}(t) \mid \mathfrak{F}_t] = \frac{\mathbf{n}(t)^2}{(n-t)(\mathbf{n}(t) - 2\mathbf{m}_2(t))} + o(1), \quad (6.4)$$

$$\mathbb{E}[\mathbf{m}_\ell(t+1) - \mathbf{m}_\ell(t) \mid \mathfrak{F}_t] = \frac{\mathbf{n}(t)^2}{(n-t)(\mathbf{n}(t) - 2\mathbf{m}_2(t))} \cdot \frac{(\ell+1)\mathbf{m}_{\ell+1}(t) - \ell\mathbf{m}_\ell(t)}{\mathbf{n}(t)} + o(1) \quad (2 \leq \ell < k), \quad (6.5)$$

$$\mathbb{E}[\mathbf{m}_k(t+1) \mid \mathfrak{F}_t] = -\frac{\mathbf{n}(t)^2}{(n-t)(\mathbf{n}(t) - 2\mathbf{m}_2(t))} \cdot \frac{k\mathbf{m}_k(t)}{\mathbf{n}(t)} + o(1). \quad (6.6)$$

Proof. **TOPROVE 31** □

Lemma 6.7 puts us in a position to derive a system of ODEs to track the random variables $\mathbf{n}(t), \mathbf{m}_2(t), \dots, \mathbf{m}_k(t)$. Specifically, we obtain the following.

Corollary 6.8. Let $\mathbf{n}, \mathbf{m}_2, \dots, \mathbf{m}_k : [0, 1] \rightarrow \mathbb{R}$ be continuously differentiable functions such that

$$\mathbf{n}(0) = 1, \quad \mathbf{m}_k(0) = \frac{d}{k}, \quad (6.7)$$

$$\frac{\partial \mathbf{n}}{\partial \theta} = -\frac{\mathbf{n}^2}{(1-\theta)(\mathbf{n} - 2\mathbf{m}_2)}, \quad (6.8)$$

$$\frac{\partial \mathbf{m}_\ell}{\partial \theta} = \frac{\mathbf{n}((\ell+1)\mathbf{m}_{\ell+1} - \ell\mathbf{m}_\ell)}{(1-\theta)(\mathbf{n} - 2\mathbf{m}_2)} \quad (2 \leq \ell < k), \quad \frac{\partial \mathbf{m}_k}{\partial \theta} = -\frac{k\mathbf{n}\mathbf{m}_k}{(1-\theta)(\mathbf{n} - 2\mathbf{m}_2)}. \quad (6.9)$$

Assume, furthermore, that

$$\sup_{\theta \in [0,1]} 2\mathbf{m}_2(\theta)/\mathbf{n}(\theta) < 1. \quad (6.10)$$

Then with probability $1 - o(n^{-2})$ for all $0 \leq t \leq n$ we have

$$\mathbf{n}(t)/n = \mathbf{n}(t/n) + o(1), \quad \mathbf{m}_\ell(t)/n = \mathbf{m}_\ell(t/n) + o(1) \quad (2 \leq \ell \leq k).$$

Proof. **TOPROVE 32** □

As a next step we construct an explicit solution to the system (6.7)–(6.9).

Lemma 6.9. *If $d < d_{\min}$, then the functions*

$$\mathbf{n}^*(\theta) = 1 - \alpha_*(\lambda(\theta)), \quad \mathbf{m}_\ell^*(\theta) = \frac{d}{k} \binom{k}{\ell} (1 - \alpha_*(\lambda(\theta)))^\ell \alpha_*(\lambda(\theta))^{k-\ell} \quad (6.11)$$

satisfy (6.7)–(6.10).

Proof. **TOPROVE 33** □

Proof. **TOPROVE 34** □

6.2.2. *Proof of Proposition 6.4.* $F'_{\text{UC},t+1}$ is the XORSAT formula that contains the variables $V'(t+1)$ that get assigned during iteration $t+1$ and the clauses $C'(t+1)$ of $F_{\text{UC},t}$ that contain variables from $V'(t+1)$ only. Also recall that $G(F'_{\text{UC},t+1})$ signifies the graph representation of this XORSAT formula. Unless $V'(t+1) = \emptyset$, the graph $G(F'_{\text{UC},t+1})$ is connected.

Lemma 6.10. *Fix $\varepsilon > 0$ and let $0 \leq t \leq (1 - \varepsilon)n$. With probability $1 - o(1/n)$ the graph $G(F'_{\text{UC},t+1})$ satisfies*

$$|E(G(F'_{\text{UC},t+1}))| \leq |V(G(F'_{\text{UC},t+1}))|.$$

Proof. **TOPROVE 35** □

Thus, with probability $1 - o(1/n)$ the graph $G(F'_{\text{UC},t+1})$ contains at most one cycle. While it is easy to check that no conflict occurs in iteration $t+1$ if $G(F'_{\text{UC},t+1})$ is acyclic, in the case that $G(F'_{\text{UC},t+1})$ contains a single cycle there is a chance of a conflict. The following definition describes the type of cycle that poses an obstacle.

Definition 6.11. *For a XORSAT formula F we call a sequence of variables and clauses $\mathcal{C} = (v_1, c_1, \dots, v_\ell, c_\ell, v_{\ell+1} = v_1)$ a toxic cycle of length ℓ if*

TOX1: c_i contains the variables x_i, x_{i+1} only, and

TOX2: the total number of negations in c_1, \dots, c_ℓ is odd iff ℓ is even.

Lemma 6.12. (i) *If $F'_{\text{UC},t+1}$ contains a toxic cycle, then a conflict occurs in iteration $t+1$.*

(ii) *If $F'_{\text{UC},t+1}$ contains no toxic cycle and $|E(G(F'_{\text{UC},t+1}))| \leq |V(G(F'_{\text{UC},t+1}))|$, then no conflict occurs in iteration $t+1$.*

Proof. **TOPROVE 36** □

Corollary 6.13. *Fix $\varepsilon > 0$ and let $0 \leq t \leq (1 - \varepsilon)n$. Then*

$$\mathbb{P}[\mathcal{R}_{t+1}] = \mathbb{P}[F'_{\text{UC},t+1} \text{ contains a toxic cycle}] + o(1/n).$$

Proof. **TOPROVE 37** □

Thus, we are left to calculate the probability that $F'_{\text{UC},t+1}$ contains a toxic cycle. To this end, we estimate the number of toxic cycles in the ‘big’ formula $F_{\text{UC},t}$. Let $T_t(\ell)$ be the number of toxic cycles of length ℓ in $F_{\text{UC},t}$.

Lemma 6.14. *Fix $\varepsilon > 0$ and let $1 \leq t \leq (1 - \varepsilon)n$.*

(i) *For any fixed ℓ , with probability $1 - O(n^{-2})$ we have*

$$\mathbb{E}[T_t(\ell) \mid \mathfrak{F}_t] = \beta_\ell + o(1), \quad \text{where } \beta_\ell = \frac{1}{4\ell} \left(d(k-1)(1 - \alpha_*) \alpha_*^{k-2} \right)^\ell = \frac{1}{4\ell} (f_n(t))^\ell.$$

(ii) *For any $1 \leq \ell \leq n$, with probability $1 - O(n^{-2})$ we have $\mathbb{E}[T_t(\ell) \mid \mathfrak{F}_t] \leq \beta_\ell \exp(\varepsilon\ell)$.*

Proof. **TOPROVE 38** □

Proof. **TOPROVE 39** □

6.2.3. *Proof of Proposition 6.5.* We combine Fact 6.2 with the tower rule. Specifically, let $0 \leq t_1 < \dots < t_h < (1 - \varepsilon)n$ be distinct time indices. Then repeated application of the tower rule gives

$$\begin{aligned} \mathbb{P} \left[\bigcap_{i=1}^h \mathcal{R}_{t_i} \right] &= \mathbb{E} \left[\prod_{i=1}^h \mathbb{1}\{\mathcal{R}_{t_i}\} \right] = \mathbb{E} \left[\mathbb{E} \left[\prod_{i=1}^h \mathbb{1}\{\mathcal{R}_{t_i}\} \mid \mathfrak{F}_{t_{i-1}} \right] \right] \\ &= \mathbb{E} \left[\left(\prod_{i=1}^{h-1} \mathbb{1}\{\mathcal{R}_{t_i}\} \right) \mathbb{P}[\mathcal{R}_{t_h} \mid \mathfrak{F}_{t_{h-1}}] \right] = \dots = \mathbb{E} \left[\prod_{i=1}^h \mathbb{P}[\mathcal{R}_{t_i} \mid \mathfrak{F}_{t_{i-1}}] \right]. \end{aligned} \quad (6.12)$$

Furthermore, Proposition 6.4 shows that with probability $1 - o(1/n)$,

$$\mathbb{P}[\mathcal{R}_{t_i} \mid \mathfrak{F}_{t_{i-1}}] = \frac{f_n(t_i)^2}{4(n - t_i)(1 - f_n(t_i))^2} + o(1/n) \quad \text{for all } 1 \leq i \leq h. \quad (6.13)$$

Combining (6.12)–(6.13) completes the proof.

6.2.4. *Proof of Proposition 6.6.* Given $\delta > 0$ pick $\varepsilon > 0$ small enough and let $t = \lceil (1 - \varepsilon)n \rceil$. We are going to show that the graph $G(\mathbf{F}_{\text{UC},t})$ is acyclic with probability at least $1 - \delta$. Since all clauses of $\mathbf{F}_{\text{UC},t}$ contain at least two variables, UCP will find a satisfying assignment if $G(\mathbf{F}_{\text{UC},t})$ is acyclic.

To show that $G(\mathbf{F}_{\text{UC},t})$ is acyclic, we observe that $\alpha_* \geq t/n$. Hence, α_* approaches one as $t/n \rightarrow 1$. Further, Fact 6.2 shows that $G(\mathbf{F}_{\text{UC},t})$ is uniformly random given the degree distribution (6.1) of the clause nodes. Indeed, the expression (6.1) shows that with probability $1 - O(n^{-2})$ the expected size of the second neighbourhood of a given variable node is asymptotically equal to

$$\gamma = \gamma(\varepsilon) = \frac{1}{(1 - \alpha_*)n} \cdot \frac{dn}{k} \sum_{\ell=2}^k \ell \binom{k}{\ell} (1 - \alpha_*)^\ell \alpha_*^{k-\ell} = d(1 - \alpha_*^{k-1}).$$

Hence, as $\lim_{\varepsilon \rightarrow 0} \gamma = 0$, the average degree of the random graph $G(\mathbf{F}_{\text{UC},t})$ tends to zero as $\varepsilon \rightarrow 0$. Therefore, for small enough $\varepsilon > 0$ the random graph $G(\mathbf{F}_{\text{UC},t})$ is acyclic with probability greater than $1 - \delta$.

6.3. **Failure of UCP for $d_{\min} < d < d_{\text{sat}}$.** In this section we assume that $d_{\min} < d < d_{\text{sat}}$. As in Section 6.2 we are going to trace UCP via the method of differential equations. In particular, we keep the notation from Section 6.2. Thus, $\mathbf{n}(t)$ signifies the number of unassigned variables after t iterations, and $\mathbf{m}_\ell(t)$ denotes the number of clauses that contain precisely $2 \leq \ell \leq k$ unassigned variables. Moreover, $\mathbf{F}_{\text{UC},t}$ is the formula comprising these variables and clauses. The following statement is the analogue of Proposition 6.3 for $d_{\min} < d < d_{\text{sat}}$. Its proof relies on similar arguments as the proof of Proposition 6.3.

Proposition 6.15. *Suppose that $d_{\min}(k) < d < d_{\text{sat}}(k)$, fix $\varepsilon, \delta > 0$ and let $0 < t < (1 - \varepsilon)\theta_* n$. Then (6.1) holds with probability $1 - O(n^{-2})$.*

Proof. **TOPROVE 40** □

Proof. **TOPROVE 41** □

ACKNOWLEDGEMENT

Amin Coja-Oghlan is supported by DFG CO 646/3, DFG CO 646/5 and DFG CO 646/6. Lena Krieg is supported by DFG CO 646/3. Maurice Rolvien is supported by DFG Research Group ADYN (FOR 2975) under grant DFG 411362735. This research was funded in part by the Austrian Science Fund (FWF) [10.55776/I6502]. For open access purposes, the authors have applied a CC BY public copyright license to any author accepted manuscript version arising from this submission.

REFERENCES

- [1] Dimitris Achlioptas and Amin Coja-Oghlan. Algorithmic barriers from phase transitions. In *Proc. 49th FOCS*, pages 793–802, 2008. doi : 10.1109/FOCS.2008.11.
- [2] Dimitris Achlioptas and Micheal Molloy. The solution space geometry of random linear equations. *Random Structures & Algorithms*, 46:197–231, 2015. doi : 10.1002/rsa.20494.
- [3] Peter Ayre, Amin Coja-Oghlan, Pu Gao, and Noëla Müller. The satisfiability threshold for random linear equations. *Combinatorica*, 40:179–235, 2020. doi : 10.1007/s00493-019-3897-3.
- [4] Béla Bollobás. *Random Graphs*. Cambridge University Press, 2001. doi : 10.1017/CB09780511814068.

- [5] Alfredo Braunstein, Marc Mézard, and Riccardo Zecchina. Survey propagation: An algorithm for satisfiability. *Random Structures & Algorithms*, 27:201–226, 2005. doi:10.1002/rsa.20057.
- [6] Amin Coja-Oghlan. A better algorithm for random k-sat. *SIAM Journal on Computing*, 39:2823–2864, 2010. doi:10.1137/09076516X.
- [7] Amin Coja-Oghlan. Belief propagation guided decimation fails on random formulas. *Journal of the ACM*, 63(49), 2017. doi:10.1145/3005398.
- [8] Amin Coja-Oghlan, Alperen A. Ergür, Pu Gao, Samuel Hetterich, and Maurice Rolvien. The rank of sparse random matrices. *Random Structures & Algorithms*, 62:68–130, 2023. doi:10.1002/rsa.21085.
- [9] Amin Coja-Oghlan, Pu Gao, Max Hahn-Klimroth, Joon Lee, Noëla Müller, and Maurice Rolvien. The full rank condition for sparse random matrices. *Combinatorics, Probability and Computing*, 33:643–707, 2024. doi:10.1017/S096354832400021X.
- [10] Amin Coja-Oghlan and Angelica Pachon-Pinzon. The decimation process in random k-sat. *SIAM Journal on Discrete Mathematics*, 26:1471–1509, 2012. doi:10.1137/110842867.
- [11] Christophe Deroulers and Rémi Monasson. Criticality and universality in the unit-propagation search rule. *European Physical Journal B*, 49:339–369, 2006. doi:10.1140/epjb/e2006-00072-6.
- [12] Olivier Dubois and Jacques Mandler. The 3-xorsat threshold. In *Proc. 43rd FOCS*, pages 769–778, 2002. doi:10.1109/SFCS.2002.1182002.
- [13] Alan Frieze and Stephen Suen. Analysis of two simple heuristics on a random instance of k-sat. *Journal of Algorithms*, 20:312–355, 1996. doi:10.1006/jagm.1996.0016.
- [14] David Gamarnik. The overlap gap property: a topological barrier to optimizing over random structures. *Proceeding of the National Academy of Sciences*, 118, 2021. doi:10.1073/pnas.2108492118.
- [15] Samuel Hetterich. Analysing survey propagation guided decimation on random formulas. In *Proc. 43rd ICALP*, number 65, 2016. doi:10.4230/LIPIcs.ICALP.2016.65.
- [16] Morteza Ibrahimi, Yash Kanoria, Matt Kranning, and Andrea Montanari. The set of solutions of random xorsat formulae. *Annals of Applied Probability*, 25:2743–2808, 2015. URL: <http://www.jstor.org/stable/24521615>.
- [17] Florent Krzakala, Andrea Montanari, Federico Ricci-Tersenghi, Guilhem Semerjian, and Lenka Zdeborová. Gibbs states and the set of solutions of random constraint satisfaction problems. *Proceeding of the National Academy of Sciences*, 104:10318–10323, 2007. doi:10.1073/pnas.0703685104.
- [18] Aude Maier, Freya Behrens, and Lenka Zdeborová. *Phys. Rev. E*, 112:014306, Jul 2025. URL: <https://link.aps.org/doi/10.1103/PhysRevE.112.014306>, doi:10.1103/PhysRevE.112.014306.
- [19] Marc Mézard and Andrea Montanari. *Information, Physics and Computation*. Oxford University Press, 2009. doi:10.1093/acprof:oso/9780198570837.001.0001.
- [20] Marc Mézard, Thierry Mora, and Riccardo Zecchina. Clustering of solutions in the random satisfiability problem. *Physical Review Letters*, 94, 2005. doi:10.1103/PhysRevLett.94.197205.
- [21] Marc Mézard, Giorgio Parisi, and Riccardo Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297:812–815, 2002. doi:10.1126/science.1073287.
- [22] Marc Mézard, Federico Ricci-Tersenghi, and Riccardo Zecchina. Two solutions to diluted p-spin models and xorsat problems. *Journal of Statistical Physics*, 111:505–533, 2003. doi:10.1023/A:1022886412117.
- [23] Michael Molloy. Cores in random hypergraphs and boolean formulas. *Random Structures & Algorithms*, 27:124–135, 2005. doi:10.1002/RSA.20061.
- [24] Boris Pittel and Gregory B. Sorkin. The satisfiability threshold for k-xorsat. *Combinatorics, Probability and Computing*, 25:236–268, 2016. doi:10.1017/S0963548315000097.
- [25] Federico Ricci-Tersenghi and Guilhem Semerjian. On the cavity method for decimated random constraint satisfaction problems and the analysis of belief propagation guided decimation algorithms. *Journal of Statistical Mechanics*, (9), 2009. doi:10.1088/1742-5468/2009/09/P09001.
- [26] Nicholas Wormald. Differential equations for random processes and random graphs. *Annals of Applied Probability*, 5:1217–1235, 1995. doi:10.1214/aoap/1177004612.
- [27] Kingsley Yung. Limits of sequential local algorithms on the random k-xorsat problem. In *Proc. 51st ICALP*, number 123, 2024. doi:10.4230/LIPIcs.ICALP.2024.123.

ARNAB CHATTERJEE, arnab.chatterjee@tu-dortmund.de, TU DORTMUND, FACULTY OF COMPUTER SCIENCE, 12 OTTO-HAHN-ST, 44227 DORTMUND, GERMANY.

AMIN COJA-OGHLAN, amin.coja-oghlan@tu-dortmund.de, TU DORTMUND, FACULTY OF COMPUTER SCIENCE AND FACULTY OF MATHEMATICS, 12 OTTO-HAHN-ST, 44227 DORTMUND, GERMANY.

MIHYUN KANG, kang@math.tugraz.at, TU GRAZ, INSTITUTE OF DISCRETE MATHEMATICS, STEYRERGASSE 30, 8010 GRAZ, AUSTRIA.

LENA KRIEG, lena.krieg@tu-dortmund.de, TU DORTMUND, FACULTY OF COMPUTER SCIENCE, 12 OTTO-HAHN-ST, 44227 DORTMUND, GERMANY.

MAURICE ROLVIER, maurice.rolvien@tu-dortmund.de, UNIVERSITY OF HAMBURG, DEPARTMENT OF INFORMATICS, VOGT-KÖLLN-STR. 30, 22527 HAMBURG, GERMANY.

GREGORY B. SORKIN, g.b.sorkin@lse.ac.uk, THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE, DEPARTMENT OF MATHEMATICS, COLUMBIA HOUSE, HOUGHTON ST, LONDON WC2A 2AE, UNITED KINGDOM