Performance of the Survey Propagation-guided decimation algorithm for the random NAE-K-SAT problem

David Gamarnik[*] Madhu Sudan[†]

Abstract

We show that the Survey Propagation guided decimation algorithm fails to find satisfying assignments on random instances of the "Not-All-Equal-K-SAT" problem, well below the satisfiability threshold. Our analysis applies to a broad class of algorithms that may be described as "sequential local algorithms" — such algorithms iteratively set variables based on some local information and/or local randomness, and then recurse on the reduced instance. Survey Propagation guided as well as Belief Propagation guided decimation algorithms, studied widely in the past, fall under this category of algorithms. Our main technical result shows that under fairly mild conditions sequential local algorithms find satisfying assignments only when the solution space is nicely connected, despite the earlier predictions by statistical physicists. Combined with the knowledge that the solution space tends to cluster well before the satisfiability threshold, our main result follows immediately. This approach of showing that local algorithms work only when the solution space is connected has been applied before in the literature: the novelty of our work is our ability to extend the approach also to sequential local algorithms.

1 Introduction

In this work we study the behavior of some "natural", statistical-physics-motivated, algorithms for constraint satisfaction problems on random instances. While these algorithms were widely studied in the past and there are many empirical evidences of success of such algorithms [MPZ02], [MM09], [BMZ05], [KMRT$^+$07], [MMW07], [CO11], there are relatively few analytical proofs, except for [CO11]. In this work we consider a large class of such algorithms and analyze their behavior on random instances of "Not-All-Equal-K-SAT (NAE-K-SAT)". We show that most of them fail to find satisfying assignments on instances with density (ratio of clauses to variables) well below the satisfiability threshold and, in particular, are incapable of breaking the so-called clustering threshold, despite the fact that in particular, the so-called Survey Propagation guided decimation algorithm was designed precisely to overcome the clustering threshold [MPZ02], [BMZ05], [MM], [KMRT$^+$07].

The precise class of algorithms we study are what we call "sequential local algorithms". Roughly, these algorithms work by assigning Boolean values to variables sequentially, with a chosen variable being assigned its value by a potentially probabilistic choice which depends on on the local neighborhood of the variable. The local neighborhood is defined to be balls of constant radius in the graph whose vertices are variables and constraints, and a variable is adjacent to a constraint if the constraint is affected by the variable. Once a variable is assigned a value, this simplifies the formula a bit (removing some constraints, and restricting others), and this in turn may influence the local neighborhoods of other variables. The algorithm updates the probabilities based on this setting and continues with its iterations till all variables are set.

[*] MIT; e-mail: gamarnik@mit.edu.Research supported by the NSF grants CMMI-1335155.

[†] Microsoft Research New England; e-mail: madhu@mit.edu

Some well-studied, though mostly empirically, classes of such sequential local algorithms are algorithms based on Belief Propagation (BP) and Survey Propagation (SP) message passing iterations. These algorithms were thus dubbed BP-guided and SP-guided decimation algorithms, the word decimation reflecting the sequential nature of these procedures. In BP-guide decimation algorithm, the local rule picks a constant sized neighborhood and a variable is assigned 1 with probability equal to the fraction of assignments where this variable is assigned the value 1 among all assignments that satisfy all clauses in this local neighborhood. The size of the neighborhood is dictated by the number iterations BP message passing iterations, which based on heuristic statistical physics considerations can be taken as constant independent of the size of the instance. SP-guided decimation algorithm uses a more complex rule for its assignments. It is based on lifting the boolean constraint satisfaction problem to a constraint satisfaction problem based on three decisions, as opposed to two decisions, but otherwise follows the same spirit.

Empirically these heuristics apparently have good performance, and often tend to find satisfying assignments with high probability primarily for constraint satisfaction problems with small number of variables per constraints, and for coloring problem with small number of colors. For some benchmark instances these algorithms outperformed all previous algorithms [MPZ02]. Further more, on the positive side, Coja-Oghlan and Panagiotou [COP12], rely on insights gained from the Survey Propagation iterations and use it to pin down the satisfiability threshold for the random “NAE-K-SAT” problem (a central constraint satisfaction problem that is also the subject of this paper and will be defined shortly) quite tightly. On the negative side, Coja-Oghlan [CO11] showed that BP-guided decimation algorithm fails to find satisfying assignments for the random K-SAT problems at densities well below the satisfiability threshold, at the densities below which even trivial algorithms succeed in finding a satisfying assignment with high probability. In this work we prove limits on an entire class of sequential local algorithms, with mild restrictions on the algorithms, for the “NAE-K-SAT” problem.

Our main proof technique relies on the clustering property of NAE-K-SAT which was earlier established for random K-SAT problem, and several other related problems, including the problem of proper coloring of sparse random graphs. Roughly speaking, the property says that, above a certain density, the Hamming distance between every pair of satisfying assignments, normalized by the number of variables, is either smaller than a certain constant δ_1 or larger than some constant $\delta_2 > \delta_1$. We then show that if a sequential local algorithm was capable of finding a satisfying assignment, then by running the algorithm twice and constructing a certain interpolation scheme, one obtains two satisfying assignments with normalized Hamming distance in the interval (δ_1, δ_2), thus obtaining a contradiction. It is precisely this clustering property that prompted the statistical physicists to design the SP-guided decimation algorithm in the first place. Thus one of the key messages of this paper is that, unfortunately, the Survey Propagation algorithm is not capable of overcoming the clustering threshold either, and we attribute its empirical success to relatively small sizes of parameter K chosen in the experiments and some clever implementation details, specifically size biasing, which we discuss briefly in the body of the paper.

The link between the clustering property and the ensuing demise of local algorithm was recently used by the authors [GS13] in a different context of optimization on random regular graphs. There the argument was used that so-called i.i.d. factor based local algorithm, are incapable of finding nearly optimal independent sets in random regular graphs, refuting an earlier conjecture by Hatami, Lovász and Szegedy [HLS]. An important technical and conceptual difference between two works is that algorithms considered in [HLS] and [GS13] are not sequential and, as a result, the analysis is much simpler. The technical difficulties arising in the present context as well as our approach to overcome them is outlined in the next next subsection. Both the present work and [GS13] establishing a fascinating link between the clustering property and hardness for local algorithms.

Random Not-All-Equal K-SAT: An instance Φ of the *Not-All-Equal K-SAT* (NAE-K-SAT) problem on n Boolean variables $x_1, \ldots, x_n$ is given by m "NAE" constraints $C_1, \ldots, C_m$ where each constraint is a set of K literals (a variable or its negation). The constraint is satisfied by a Boolean assignment if not all literals in the clause take on the same value. Φ is satisfied if all constraints are satisfied.

In this work we will be interested in the behavior of sequential local algorithms on *random* instances of NAE-K-SAT. A random instance is chosen by picking each clause independently and uniformly among the set of all possible clauses on K literals. The ratio $d = m/n$, known as the density of the problem, is a basic parameter. It is known [Fri99] that there is a sharp threshold d_n such that as d increases and passes through d_n the probability of satisfiability drops from nearly one to nearly zero. It is conjectured that d_n converges to a limit d_c, but this convergence remains a major open problem. The best estimate on d_n right now is from the work of Coja-Oghlan and Panagiotou [COP12], who show that $d_n \approx d_s \triangleq 2^{K-1} \ln 2 - \ln 2/2 - 1/4 - o_K(1)$, where $o_K(1)$ is a function converging to zero as $K \to \infty$. To be precise, denoting by $d_{s,K,*}$ the supremum of d such that random instance with density d is satisfiable with probability approaching one, as $n \to \infty$, and denoting by $d^*_{s,K}$ the infimum of d such that the random instance is not satisfiable for density d, it is the case that both $d_{s,K,*}$ and $d^*_{s,K}$ are $2^{K-1} \ln 2 - \ln 2/2 - 1/4 - o_K(1)$. We refer to d_s as the *satisfiability threshold*, despite the ambiguity involving term $o_K(1)$. The proof methods used in [COP12] and similar earlier bounds employ existential arguments and thus the question is finding algorithms which find satisfying assignment in random instances when $d < d_s$.

Sequential Local Algorithms To describe our results, we first need to define our notion of sequential local algorithms. We do informally here, and then formally in Section 2. A sequential local algorithm starts with an input which is an instance of NAE-K-SAT and assigns variables iteratively to 0 or 1 based on some local rule. Of course, once a variable is assigned a value 0 or 1, the resulting instance is no longer an NAE-K-SAT instance. E.g., in the constraint $NAE(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$ if we set $x_3 = 1$, we are left with the constraint $(\neg x_1 \vee \neg x_2)$ which is a simple 2-CNF clause. So the class of instances that a local algorithm needs to work with is called a *reduced instances* of NAE-K-SAT, and includes three types of constraints: (1) (original/neutral) length K constraint on K literals requiring them to be not-all-equal, (2) *positive* constraints on $< K$ literals requiring at least one to be 1, and (3) *negative* constraints on $< K$ literals requiring at least one to be 0.

Given a reduced instance of NAE-K-SAT, and possibly some local randomness, the local rule $\tau(x_i)$ assigns to variable x_i a real number between 0 and 1 which determines the probability with which x_i would be set to 1, *if* the local rule were to be activated now. But only one rule is activated at a time, i.e., x_i is set to 1 with probability $\tau(x_i)$ for one choice of i, and then the instance reduced by the choice, leading possibly to new probabilities from the local function τ. Locality of τ simply implies that $\tau(x_i)$ does not depend on constraints that are far away from x_i in the constraint-variable adjacency graph.

Finally, one also needs to determine the order in which variables are selected to be set to 0 or 1. In this work we consider the case where this ordering is random. We note that the literature does include cases where the ordering is a function of the $\tau(\cdot)$ values (more on this in Subsection 2.3), even though there has been no justification provided for this choice. We are unable to analyze such a "non-local" rule for selection of variable order. Thus for our purposes, the local rule τ uniquely specifies the sequential local algorithm which we denote A_τ. By $A_\tau(\Phi)$ we denote the assignment (which may be a random variable) returned by A_τ on input Φ.

Our results. In order to describe our results we need one more notion, that of a *balanced* local rule. For this notion, in turn we need the notion of a complementary instance. The complement $\overline{C}$ of a neutral constraint C is C itself. The complement of a positive constraint $(C, +)$ is the negative constraint $(C, -)$ and vice versa. The complement $\overline{\Phi}$ of a reduced instance of NAE-K-SAT is the instance complementing every constraint in Φ. Note that the complementary constraint $\overline{\Phi}$ is also a reduced instance of NAE-K-SAT and assignment x satisfies Φ if and only if its complementary assignment $1-x$ satisfies $\overline{\Phi}$.

We say that a local rule is *balanced* if $\tau(x_i, \Phi) = 1 - \tau(x_i, \overline{\Phi})$ for every reduced instance Φ of NAE-K-SAT. Note that in particular this implies that for unreduced instances Φ, $\tau(x_i, \Phi) = 1/2$ for every i. Our main theorem, stated formally in Section 2 (see Theorem 2.4), shows that balanced sequential local algorithms fail to find satisfying assignments (with overwhelming probability) in instances of density that is a constant factor smaller than the satisfiability threshold.

Techniques and comparison with recent results. Our proof follows a fairly simple outline which we describe first. We exploit a "clustering" phenomenon that is by now well-understood for most constraint satisfaction problems. This phenomenon studies the "geometry" of the solution space of instances of NAE-K-SAT in the following sense: Given an instance Φ, let $S \subseteq \{0,1\}^n$ be all satisfying assignments of Φ and put an edge between $x, y \in S$ if (informally), $\rho(x,y)$, the Hamming distance between x and y, is $o(n)$. The clustering phenomenon asserts that as one increases the density of an instance of the constraint satisfaction problem from say 0 to the satisfiability threshold d_s, the solution space goes from being one large connected component, to a collection of smaller clusters which slowly diminish in size, till all (or almost all) the clusters become empty.

It has often been suggested that this clustering phenomenon is also a barrier to computational complexity. Indeed in [CO11], one sees a coincidence of the thresholds at which belief propagation fails to work, and where the clustering starts. Till recently though, there was no formal reduction shown between the clustering phenomenon and the failure of common algorithms.

The first steps towards a formal connection was established by the authors [GS13] where it is shown that clustering leads to a failure of local algorithms. Their work shows that one can run a local algorithm twice, with coupled randomness to produce two different satisfying assignments that are at an intermediate level of overlap, something that would contradict an appropriate phrasing of the clustering phenomenon. While their work is suggestive that the failure may be evidence of a broader phenomenon, their work did not cover natural algorithms like belief propagation or survey propagation. Specifically their work covers local algorithms, but not sequential local algorithms.

Our work extends the previous work to establish a formal connection in these important cases. There are two technical hurdles that needed to be overcome to make this extension possible. To show that their "coupling argument" works, the previous work needed to ensure that if the local algorithm is run twice, on independent randomness, then it will produce two solutions of very small overlap. Of course, such an argument can be true only if the local algorithm is randomized, and for algorithms working on d-regular graphs it is easy to argue that the algorithm better be randomized (since almost surely there is no local distinctions). In our case, there is a lot of useful information in local neighborhoods and this could potentially be used by algorithms to find satisfying assignments. We overcome this hurdle by proper abstraction: We notice that the commonly employed local algorithms are randomized, and indeed even balanced so a priori, the chance of assigning 1 to a variable is exactly $1/2$. (Of course, this is where the fact that we work with NAE-K-SAT helps us.) And for balanced local rules, we immediately manage to find that solutions derived from independent randomness end up being reasonably far from each other.

Having established that there are two random strings U and $\tilde{U}$ such that the assignments produced by the algorithm on randomness U and $\tilde{U}$, say x and $\tilde{x}$ have small overlap, previous works used a continuity/hybrid argument to show that if the algorithm is run with the string Z that equals U on any coordinate with probability p and $\tilde{U}$ with probability $1-p$, then the assignment essentially varies continuously with p leading eventually to some assignment z that has the forbidden amount of overlap with x (as forbidden by the clustering phenomenon).

Much of this argument turns out to be too complex to repeat in our setting. First the continuity argument relied heavily on the fact that rules were completely local, i.e., determined by a constant number of variables, and in the sequential case this is not true. Next it is hard to determine explicitly how the assignment changes when some random variable is perturbed. We overcome these constraints by first serving up an argument showing that even in the sequential setting the long chains of influence are not too long (specifically have length $O(\ln n/\ln\ln n)$) — this argument uses a reasoning that has been seen before in several other "local" contexts, that chains of length ℓ have probability roughly $1/\ell!$ of being relevant, while there are only singly exponentially many chains of length ℓ in any graph. (See [GG10],[NO08], for use of this reasoning in different contexts.) In turn this tells us that any single random variable only influences $o(n)$ variables even in a sequential algorithm. Finally we end up using a much simpler argument than the earlier continuity to show that this "o(n)-influence" feature is sufficient to find assignments with a forbidden overlap.

We remark that while much of the argument is a simplification of previous arguments, this simplification enables us to reach much more important classes of algorithms. In particular it reproduces (at least in the case of NAE-K-SAT) the main theorem of Coja-Oghlan [CO11] with a much simpler proof. More importantly it is the first rigorous result on the performance of the SP-guided decimation algorithm for random constraint satisfaction problems.

While the present work is only devoted to random NAE-K-SAT, it appears that our approach is applicable to other models with appropriate symmetry, such as the problem of proper coloring of random graphs. Establishing limits of sequential local algorithms for this model is an ongoing work. The random K-SAT model, however, does present some difficulties since there is obvious symmetry in the problem leading to a "minimum amount of randomness" of the underlying local algorithm. One potential approach is to employ the randomness implied by the details of the Survey Propagation iteration. This would be an interesting future research.

Finally we conclude by stressing that while our algorithm captures the cleaner versions of these natural algorithms (where the variable ordering is determined by clean rules), more sophisticated implementations are common and it would be interesting to see if these other implementations suffer from the same limitations, or if they provide the key to overcome the limitations. Thus far the literature has not stressed the impacts of these "lesser" choices, and at the very least our work could divert attention towards these.

Organization. The remainder of the paper is organized as follows. The NAE-K-SAT model, the formal description of a sequential local algorithm, the statement of the main result and applications to the BP-guided and SP-guided decimation algorithms are the subject of the next section. Some preliminary technical results are established in Section 3. In particular, we establish bounds on the influence range of variables. The proof of the main result is in Section 5. The proofs of some of the technical results are delayed till the Appendix section.

2 Formal statement of main result

In this section we formally present our main result. Before doing so we first introduce the mathematical notation and preliminaries needed to state our result.

2.1 Not-All-Equal-K-Satisfiability (NAE-K-SAT) problem

At the expense of being redundant, let us recall the NAE-K-SAT problem. An instance Φ, of NAE-K-SAT problem is described as a collection of n binary variables $x_1, \ldots, x_n$ taking values 0 and 1 and a collection of m constraints $C_1, \ldots, C_m$ where each constraint is given by a subset of K literals. Each literal is a variable x in $x_1, \ldots, x_n$ or negation $\bar{x}$ of a variable. We will often use the phrase "clause" as a synonym for constraint. An assignment is a function $\sigma : \{x_1, \ldots, x_n\} \to \{0, 1\}$. σ satisfies Φ if in every clause, we have one literal valued 1 and one literal valued 0. For every assignment $\sigma = (\sigma(x_i), 1 \leq i \leq n)$, let $\bar{\sigma} = 1 - \sigma$ be the assignment given by $\bar{\sigma}(x_i) = 1 - \sigma(x_i), 1 \leq i \leq n$. Given a formula Φ, denote by $\mathbb{SAT}(\Phi) \subset \{0, 1\}^n$ the (possibly empty) set of satisfying assignments σ. The following "complementation closure" and resulting "balance" property of NAE-K-SAT are immediate (and do not hold for the K-SAT problem)

Observation 2.1. *For every instance Φ of the NAE-K-SAT problem and assignment σ, we have that σ satisfies Φ if and only if $\bar{\sigma}$ satisfies Φ. Consequently, suppose $\mathbb{SAT}(\Phi) \neq \emptyset$. Then if σ is drawn uniformly from $\mathbb{SAT}(\Phi)$, then for every $1 \leq i \leq n$ we have*

$$\mathbb{P}(\sigma(x_i) = 0) = \mathbb{P}(\sigma(x_i) = 1) = 1/2.$$

Reduced Instances. We now introduce some notations for *reduced* instances of NAE-K-SAT. A clause of a reduced instance C is given by a set of at most K literals, along with a sign $\text{sign}(C) \in \{+, -, 0\}$. Furthermore, C has exactly K literals if and only if $\text{sign}(C) = 0$. (Sometimes we refer to these signs as decorations.) An assignment σ satisfies a reduced clause C if one of the following occurs: $\text{sign}(C) = +$ and some literal in C is assigned 0 by σ, OR $\text{sign}(C) = -$ and some literal in C is assigned 1 by σ, OR $\text{sign}(C) = 0$ and there is at least one 0 literal and one 1 literal in C under the assignment σ. A reduced NAE-K-SAT instance Φ consists of one or more reduced clauses, and σ satisfies Φ if it satisfies all clauses in Φ.

Note that Observation 2.1 does not necessarily hold for the reduced instances of NAE-K-SAT problem. Instances in which every clause has sign 0 will be called non-reduced instances.

Complements Given a clause C in a reduced instance of NAE-K-SAT, its *complement*, denoted $\bar{C}$, is the clause with the same set of literals, and its sign being flipped — so if $\text{sign}(C) = +$ then $\text{sign}(\bar{C}) = -$, if $\text{sign}(C) = -$ then $\text{sign}(\bar{C}) = +$, and if $\text{sign}(C) = 0$ then $\text{sign}(\bar{C}) = 0$. Given a reduced instance Φ of NAE-K-SAT, its *complement* $\bar{\Phi}$ is the instance with the complements of clauses of Φ.

We now make the following observation, whose proof is immediate.

Observation 2.2. *Given reduced instances Φ on variables $x_1, \ldots, x_n$ and Ψ on variables $x_1, \ldots, x_{n+t}$ suppose Φ is the instance derived by reducing Ψ with the assignment $\sigma : \{x_{n+1}, \ldots, x_{n+t}\}$. Then $\bar{\Phi}$ is the reduced instance obtained by reducing $\bar{\Psi}$ with the assignment $\bar{\sigma}$, where $\bar{\sigma}(x_i) = 1 - \sigma(x_i)$.*

Namely, whenever a reduced formula Φ is obtained from a non-reduced formula $\tilde{\Phi}$ by setting some variables of Ψ, setting the same variables to opposite values generates the complement of Ψ.

Random NAE-K-SAT problem We denote by $\mathbf{\Phi}(n, dn)$ a random (non-reduced) instance of NAE-K-SAT problem on variables $x_1, \ldots, x_n$ and $\lfloor dn \rfloor$ clauses $C_1, \ldots, C_m$ generated as follows. The variables in each clause C_j are chosen from $x_1, \ldots, x_n$ uniformly at random without replacement, independently for all $j = 1, 2, \ldots, m$. Furthermore, each x variable is negated (namely appears as $\bar{x}$) with probability 1/2 independently for all variables in the clause and for all clauses. We are interested in the regime when $n \to \infty$ and d is constant, which we refer to as *density* of clauses to variables.

Graphs associated with NAE-K-SAT instances. Two graphs related to an instance Φ of the NAE-K-SAT problem are important to us. The first is the so-called *factor graph*, denoted $\mathbb{F}(\Phi)$, which is a bipartite graphs with left nodes corresponding to the variables and right nodes to the clauses and a clause node has edges to all variables in its set. The edges are labelled positive or negative to indicate the polarity of the literal in the clause. In case of reduced NAE-K-SAT instances, clause vertices are also labelled with the sign of the clause. Thus the factor graph may be viewed as a representation of the NAE-K-SAT instance.

The second graph we associate with Φ is the *variable-to-variable* graph of Φ, denoted $\mathbb{G}(\Phi)$, which has nodes corresponding to the variables and two nodes are adjacent if they appear in some clause together. Note that in contrast to the factor graph, the variable-to-variable graph loses information about the NAE-K-SAT instance Φ.

Local neighborhoods Given a (possibly reduced) instance Φ of a NAE-K-SAT problem, a variable x in this instance, and an even integer $r \geq 1$, we denote by $B_\Phi(x, r)$ the corresponding depth-r neighborhood of x in $\mathbb{F}(\Phi)$, the factor graph of Φ. When the underlying formula Φ is unambiguous, we simply write $B(x, r)$. We restrict r to be even so that for every clause appearing in $B(x, r)$ all of its associated variables also appear in $B(x, r)$. Abusing notation slightly we also use $B(x, r)$ to denote the reduced instance of NAE-K-SAT induced by the clauses in $B(x, r)$. Since r is even we have that the factor graph of this induced instance is $B(x, r)$. In light of this, observe that $B(x, r)$ is also a reduced instance of a NAE-K-SAT problem.

2.2 Sequential local algorithms for NAE-K-SAT problem and the main result

We now define the notion of sequential local algorithms formally and describe our main result.

Fix a positive even integer $r \geq 0$. Denote by $\mathcal{SAT}_r$ the set of all NAE-K-SAT reduced and non-reduced instances Φ with a designated (root) variable x such that the distance from x to any other variable in Φ is at most r in $\mathbb{F}(\Phi)$.

Given any function $\tau : \mathcal{SAT}_r \to [0, 1]$ which takes as an argument an arbitrary member $(H, x) \in \mathcal{SAT}_r$ and outputs a value (probability) in $[0, 1]$, we describe below a sequential local algorithm, which we refer to as the *τ-decimation algorithm*, for solving NAE-K-SAT problem. Given a positive even integer r, the depth-r neighborhood $B(x_i, r) = B_{\mathbf{\Phi}(n,dn)}(x_i, r)$ of any fixed variable $x_i \in [n]$ in the formula $\mathbf{\Phi}(n, dn)$ is a valid argument of the function τ, when the root of the instance $B(x_i, r)$ is assigned to be x_i. This remains the case when some of the variables $x_1, \dots, x_n$ are set to particular values and all of the satisfied and violated clauses are removed. In this case $B(x_i, r)$ is a reduced instance. In either case, the value $\tau(B(x_i, r))$ is well defined for every variable x_i which is not set yet. The value $\tau(B(x_i, r))$ is intended to represent the probability with which the variable x_i is set to take value 1 when its neighborhood is a reduced or non-reduced instance $B(x_i, r)$, according to the local algorithm. Specifically, we now describe how the function τ is used as a basis of a local algorithm to generate an assignment $\sigma : \{x_1, \dots, x_n\} \to \{0, 1\}$.

τ-decimation algorithm

```
INPUT:
an instance $\Phi$ of a NAE-K-SAT formula on binary variables $x_1, \dots, x_n$,
a positive even integer $r$,
function $\tau$.

Set $\Phi_0 = \Phi$.
FOR $i = 1 : n$
Set $\sigma(x_i) = 1$ with probability $\tau(B_{\Phi_{i-1}}(x_i, r))$
Set $\sigma(x_i) = 0$ with the remaining probability $1 - \tau(B_{\Phi_{i-1}}(x_i, r))$.
Set $\Phi_i$ to be the reduced instance obtained from $\Phi_{i-1}$ by fixing the value of $x_i$ as above, removing satisfied and violated clauses and decorating newly generated partially satisfied constraints with $+$ and $-$ appropriately.
```

OUTPUT $\sigma(x_1),\dots,\sigma(x_n)$.

In particular, even if at some point a contradiction is reached and one of the clauses is violated, the algorithm does not stop but proceeds after the removing violated clauses from the formula. This is assumed for convenience so that the output $\sigma(x_i)$ is well defined for all variables $x_i, 1 \le i \le n$ even if the assignment turns out to be not satisfying. We denote by $\sigma_{\Phi,\tau}$ the (random) output $\sigma(x_1),\dots,\sigma(x_n)$ produced by the τ-decimation algorithm above. We say that τ-decimation algorithm solves instance Φ if the output $\sigma_{\Phi,\tau}$ is a satisfying assignment, namely $\sigma_{\Phi,\tau} \in \mathbb{SAT}(\Phi)$.

We now formally define the following important symmetry condition.

Definition 2.3. *We say that a local rule $\tau : \mathcal{SAT}_r \to [0,1]$ is* balanced *if for every instance $\Phi \in \mathcal{SAT}_r$, we have $\tau(\Phi) = 1 - \tau(\bar{\Phi})$.*

The balance condition above basically says that the τ-decimation algorithm does not have a prior bias in setting variables to 1 vs 0. In particular, when the instance is non-reduced, τ-decimation algorithm sets variable values equi-probably, consistently with Observation 2.1. This condition will allow us to take advantage of Observation 2.2 when applying the rule τ to reduced instances.

We now state the main result of the paper.

Theorem 2.4. *There exists K_0 such that for every $K \ge K_0$, $d > 2^{K-2}\ln 2$, $r > 0$ and every balanced local rule $\tau : \mathcal{SAT}_r \to [0,1]$ the following holds:*

$$\lim_{n\to\infty} \mathbb{P}(\sigma_{\mathbf{\Phi}(n,dn),\tau} \in \mathbb{SAT}(\mathbf{\Phi}(n,dn))) = 0.$$

Namely, with overwhelming probability, τ-decimation algorithm fails to find a satisfying assignment. As we have mentioned above, the threshold for satisfiability is $2^{K-1}\ln 2 - \ln 2/2 - 1/4 - o_K(1)$. Thus our theorem implies that sequential local algorithms fail to find a satisfying assignment at densities approximately half of the satisfiability threshold.

2.3 BP-guided and SP-guided decimation algorithms as local sequential algorithms

We now show that the decimation versions of Belief Propagation (BP) and its extended version Survey Propagation (SP) algorithm, considered in many prior papers are in fact special cases of τ-decimation algorithms as described in the previous section. As a consequence we have that the negative result described in Theorem 2.4 applies to these algorithms as well.

The BP and SP algorithms are designed to compute certain marginal values associated with a NAE-K-SAT instance Φ and reduced instances obtained after some of the variables are set. The natural interpretation of these marginals is that variables may be set according to these marginals sequentially, while refining the marginals as decisions are made. It is common to call such algorithms BP-guided decimation algorithm and SP-guided decimation algorithms. We now describe these algorithms in detail, starting from the BP and BP-guided decimation algorithms.

Belief Propagation. The BP algorithm is a particular message-passing type algorithm based on variables and clauses exchanging messages on the bi-partite factor graph $\mathbb{F}(\Phi(n,dn))$. After several rounds of such exchange of messages, the messages are combined in a specific way to compute marginal probabilities.

However, the relevant part for us is the fact that if the messages are passed only a constant r number of rounds, then for every variable x_i such that the neighborhood $B(x_i, r)$ is a tree in the original factor graph, the computed marginals $\mu(x_i)$ are precisely the ratio of the number of assignments satisfying NAE-K-SAT formula $B(x_i, r)$ which set x_i to one to the number of such assignments which set this variable to zero. In fact for the majority of variables indeed $B(x_i, r)$ is a tree indeed - a well-known fact for the random formula $\mathbf{\Phi}(n, dn)$. Thus most of the times BP iterations compute marginal values corresponding to the ratio described above. These marginals are then used to design the BP-guided decimation algorithm as follows. Variable x_1 is selected and BP algorithm is used to compute its marginal $\mu(x_1)$ with respect to the neighborhood tree $B(x_1, r)$. Then the decision $\sigma(x_1)$ for this variable is set to $\sigma(x_1) = 1$ with probability $\mu(x_1)/(\mu(x_1) + 1)$ and $\sigma(x_1) = 0$ with probability $1/(\mu(x_1) + 1)$. Namely, the variable is set probabilistically proportionally to the ratio of solutions setting it to zero vs setting it to one. Then for the reduced formula on variables $x_2, \ldots, x_n$, the variable x_2 is selected. The marginal $\mu(x_2)$ with respect to the neighborhood $B(x_2, r)$ for this reduced formula is computed and the value $\sigma(x_2)$ is determined based on $\mu(x_2)$ similarly, and so on. The procedure is called BP-guided decimation algorithm. It is thus parametrized by the computation depth r.

It is now clear the such a BP-guided decimation algorithm is precisely the τ-decimation algorithm where $\tau(B(x_i, r)) = \mu(x_i)/(\mu(x_i) + 1)$ - the marginal probability of the variable corresponding to the reduced formula $B(x_i, r)$. Furthermore, such τ rule satisfies the balance condition described in Definition 2.3. Thus, as an implication of our main result, Theorem 2.4, we conclude that BP-guided decimation algorithm fails to find a satisfying assignment for $\Phi(n, dn)$ in the regime where our result on τ-decimation algorithms applies:

Corollary 2.5. *There exists K_0 such that for every $K \geq K_0$, $d > 2^{K-2} \ln 2$ and $r > 0$*

$$\lim_{n\to\infty} \mathbb{P}(\textit{BP-guided decimation algorithm solves } \mathbf{\Phi}(n, dn)) = 0.$$

Our result parallels a similar result by Coja-Oghlan [CO11] for random K-SAT problem, although his result is achieved using a much more sophisticated technique and, unlike our result, does not rely directly on the clustering property underlying Theorem 4.1. On the other hand, his result applies to densities $d > \rho(2^K/K)$ for some constant ρ independent from K, which is well below the density $2^{K-1} \ln 2$ - the appropriate analogue of our threshold $2^{K-2} \ln 2$ for the NAE-K-SAT version.

We should also note, that in the experimental results reported in the literature, the BP-guided decimation algorithm is in fact conducted in a size-biased way in the following sense. Instead of fixing variables in order $x_1, \ldots, x_n$, at each step the algorithm first sorts all the remaining variables according to their marginals μ and sets the assignment $\sigma(x_k)$ corresponding to the variable with the highest bias of the marginal. Namely it finds $x_k = \arg\max_i(\mu(x_i), \mu^{-1}(x_i))$, and sets $\sigma(x_k)$ to one with probability $\mu(x_k)/(\mu(x_k) + 1)$ and zero with the remaining probability, where the range of i is the set of remaining variables. However, it appears that there is no reasoning based on the statistical physics theory which claims that presorting the variables according to the marginals is a crucial step for the BP-guided decimation to succeed. Size biasing rather appears to be a sensible implementation detail of the algorithm. Another difference between our description of BP-guided decimation algorithm and the way it was implemented in experimental result is that BP iterations are run till the convergence (up to a certain tolerance) is reached, as opposed to running the iterations for a certain fixed number of steps r, as is suggested above. However, the reasoning based on statistical physics suggest that the convergence should take place in fact exponentially fast in r and thus BP computations based on a fixed number of rounds should work as well. Notably, as in our case, the analysis by Coja-Oghlan applies only the case of BP algorithm running for constantly many rounds (bounded depth), and decimation conducted in non-size-biased way. Thus our result indeed extends the result by Coja-Oghlan to the case of NAE-K-SAT problem. While it would be interesting to extend our result to the case when the decimation is done in a size-biased way, it is not clear how to pose appropriately the question of extending our result to the case when BP computations are done till convergence, since one first has to establish that such a convergence takes place.

In Appendix A we describe Survey Propagation in similar detail. The algorithm is signficantly more complex to describe, but we show once again that its decimation version is a τ-decimation algorithm and that τ is a balanced rule. As a consquence we conclude that SP-guided decimation algorithm also fails to find satisfying assignments for instances with density larger than $(1/2)d_s$:

Corollary 2.6. *There exists K_0 such that for every $K \geq K_0$, $d > 2^{K-2}\ln 2$ and $r > 0$*

$$\lim_{n\to\infty} \mathbb{P}(\textit{SP-guided decimation algorithm solves } \mathbf{\Phi}(n, dn)) = 0.$$

3 Local algorithms and long-range independence

In this section we obtain some preliminary results needed for the proof of our main result, Theorem 2.4. Specifically we prove two structural results about the τ-decimation algorithm for a local rule τ.

The first is simple to state - we show that balanced local rules lead to unbiased decisions, for *every* un-reduced NAE-K-SAT instance: specifically the marginal probability that a variable is set to 1 is $1/2$. More generally we show that the probability that a variable is set to 1 in any instance Φ equals the probability that the same variable is set to 0 in the complementary instance $\bar{\Phi}$. (See Lemma 3.1.) This lemma later allows us to find satisfying assignments with small overlap in random instances $\Phi(n, dn)$.

Next, we consider the "influence" of a decision $\sigma(x_i) \in \{0, 1\}$ and ask how many other variables are affected by this decision. In particular, we show that the decisions σ assigned to a pair of fixed variables x_i and x_j are asymptotically independent as $n \to \infty$. Namely, the decisions exhibit a long-range independence. Such a long-range independence is not a priori obvious, since setting a value of a variable x_i can have a downstream implications for setting variables $x_j, j \geq i$. We will show, however, that the chain of implications, appropriately defined is typically short. Definition 3.2 and Proposition 3.4 formalize these claims.

In what follows, we first introduce some notation that makes the decisions of the randomized algorithm more formal and precise. We then prove the two main claims above in the following subsections.

3.1 Formalizing random choices of a τ-decimation algorithm

The τ-decimation algorithm described in the previous section is based on the ordering of the variables x_i, since the values $\sigma(x_i)$ are set in the order $i = 1, 2, \ldots, n$. In the case of the random NAE-K-SAT formula $\mathbf{\Phi}(n, dn)$, due to symmetry we may assume, without the loss of generality, that the ordering is achieved by assigning random i.i.d. labels chosen uniformly from $[0, 1]$ and using order statistics for ordering of variables. (This is equivalent to renaming the variables at random and this renaming will be convenient for us.) Specifically, let $\mathbf{Z} = (Z_i, 1 \leq i \leq n)$ be the i.i.d. sequence of random variables with uniform in $[0, 1]$ distribution. Let $\pi : [n] \to [n]$ be the permutation given by the order statistics of $\mathbf{Z}$. Namely $Z_{\pi(1)} > Z_{\pi(2)} > \cdots > Z_{\pi(n)}$. Then we assume that when the τ-decimation algorithm is performed, the first variable selected is $x_{\pi(1)}$ (as opposed to x_1), the second variable selected is $x_{\pi(2)}$ (as opposed to x_2), etc. From now on we assume that τ-decimation algorithm performed on a random instance of the NAE-K-SAT problem $\mathbf{\Phi}(n, dn)$ is done according to this ordering.

To facilitate the randomization involved in selecting randomized decisions based on the τ rule, consider another i.i.d. sequence $\mathbf{U} = (U_i, 1 \leq i \leq n)$ of random variables with the uniform in $[0, 1]$ distribution, which is independent from the randomness of the instance $\mathbf{\Phi}$ and sequence $\mathbf{Z}$. The purpose of the sequence is to serve as random seeds for the decision $\sigma(x_i)$ based on τ. Specifically, when the value $\sigma(x_i)$ associated with variable x_i is determined, it is done so according to the rule $\sigma(x_i) = 1$ if $U_i < \tau(B(x_i, r))$ and $\sigma(x_i) = 0$ otherwise, where $B(x_i, r) = B_{\Phi_{i-1}}(x_i, r)$ is the reduced NAE-K-SAT instance rooted at x_i, observed at a time when the decision for x_i needs to be made. Namely, the τ-algorithm is faithfully executed. Conditioned on $\mathbf{Z}, \mathbf{U}$ and $\mathbf{\Phi}$, the output $\sigma : [n] \to \{0, 1\}$ is uniquely determined. We denote by $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_i), 1 \le i \le n$ the output of the τ-algorithm conditioned on the realizations $\Phi, \mathbf{z}, \mathbf{u}$ of the random instance $\mathbf{\Phi}(n, dn)$, vector $\mathbf{Z}$ and vector $\mathbf{U}$, respectively. Similarly, we denote by $B_{\Phi,\mathbf{z},\mathbf{u}}(x_i, r), 1 \le i \le n$ the (possibly) reduced NAE-K-SAT instance corresponding to the r-depth neighborhood of variable x_i at the time when the value of x_i is determined by the τ-decimation algorithm. In particular, $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_i) = 1$ if $u_i \in [0, \tau(B_{\Phi,\mathbf{z},\mathbf{u}}(x_i, r))]$ and $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_i) = 0$ if $u_i \in (\tau(B_{\Phi,\mathbf{z},\mathbf{u}}(x_i, r)), 1]$.

3.2 Implications of balance

We now establish the following implication of the the Definition 2.3 of balanced local rules. Madhu's Note: There should a reference to this lemma in Section 5 when we say that expected overlap between σ^1 and σ^2 is $n/2$.

Lemma 3.1. *For every formula Φ, and vectors $\mathbf{z}, \mathbf{u}$, the following identities hold for every variable $x_i, 1 \le i \le n$:*

$$B_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_i, r) = \bar{B}_{\Phi,\mathbf{z},\mathbf{u}}(x_i, r), \tag{1}$$

$$\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_i) = 1 - \sigma_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_i), \tag{2}$$

where $\bar{\mathbf{u}}$ is defined by $\bar{u}_i = 1 - u_i, 1 \le i \le n$. As a result, when $\mathbf{U}$ is a vector of i.i.d. random variables chosen uniformly from $[0, 1]$, for Φ and $\mathbf{z}$, the following holds for all $i = 1, 2, \ldots, n$:

$$\mathbb{P}(\sigma_{\Phi,\mathbf{z},\mathbf{U}}(x_i) = 0) = 1/2. \tag{3}$$

Note, that the randomness in the probability above is with respect to $\mathbf{U}$ only and the claim holds for every formula Φ and every vector $\mathbf{z}$.

Proof. We prove the claim by induction on $x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)}$, where π is the permutation generated by $\mathbf{z}$, that is $z_{\pi(1)} > z_{\pi(2)} > \cdots > z_{\pi(n)}$. Specifically, we will show by induction that for every $i = 0, 1, 2, \ldots, n$, just before the value of variable $x_{\pi(i)}$ is determined, the identity (2) holds for all variables $x_{\pi(j)}, j \le i-1$ (namely for variables for which the value is already determined at time i), and the identity (1) in fact holds for all neighborhoods $B_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(k)}, r), i \le k \le n$ and $B_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_{\pi(k)}, r), i \le k \le n$, and not just for $B_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)}, r)$ and $B_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_{\pi(i)}, r)$.

For the base of the induction corresponding to $i = 1$, no variables are set yet and all the neighborhoods $B_{\Phi,\mathbf{z},\mathbf{u}}(x_k, r), B_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_k, r), 1 \le k \le n$ correspond to non-reduced instances, for which by our convention, its symmetric complement is the instance itself. Namely $B_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_k, r) = \bar{B}_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_k, r) = B_{\Phi,\mathbf{z},\mathbf{u}}(x_k, r)$, and thus (1) is verified.

Fix $i \ge 1$ and assume now the inductive hypothesis holds for $j \le i$. In particular, the values $\sigma(x_{\pi(j)})$ are determined for $j = 1, \ldots, i-1$ under $\mathbf{u}$ and $\bar{\mathbf{u}}$. Now consider the step of assigning the value of $x_{\pi(i)}$. We have $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)}) = 1$ iff $u_{\pi(i)} \le \tau(B_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)}, r))$ and $\sigma_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_{\pi(i)}) = 1$ iff $\bar{u}_{\pi(i)} \le \tau(B_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_{\pi(i)}, r))$. By the inductive assumption we have that $B_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_{\pi(i)}, r) = \bar{B}_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)}, r)$. By Observation 2.2 we have $\tau(\bar{B}_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)}, r)) = 1 - \tau(B_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)}, r))$. Since $\bar{u} = 1 - u$, we conclude that $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)}) = 1$ iff $\sigma_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_{\pi(i)}) = 0$ and vice verse. Namely, $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)}) = 1 - \sigma_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_{\pi(i)})$ and identity (2) is verified.

It remains to show that identity (1) still holds for all variables after the value $\sigma(x_{\pi(i)})$ is determined. All neighborhoods $B(x_k, r)$ which do not contain $x_{\pi(i)}$ are not affected by fixing the value of $x_{\pi(i)}$ and thus the identity holds by the inductive assumption. Suppose $B(x_k, r)$ contains $x_{\pi(i)}$. This means this neighborhood contains one or several clauses which contains $x_{\pi(i)}$. Fix any such clause C. If this clause was unsigned under $\mathbf{u}$, then by the inductive assumption it was also unsigned under $\bar{\mathbf{u}}$ (as the instances under $\mathbf{u}$ and $\bar{\mathbf{u}}$ are complements of each other). The clause then becomes signed after fixing the value of $x_{\pi(i)}$, and, furthermore, the signs will be opposite under $\mathbf{u}$ and $\bar{\mathbf{u}}$, since (2) holds for $x_{\pi(i)}$ as we have just established.

Now suppose the clause was signed $+$ under $\mathbf{u}$. Then again by the inductive assumption it was signed $-$ under $\bar{\mathbf{u}}$. In this case if the assignment $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)})$ satisfies C, then the clause remains signed $+$ after setting the value of $x_{\pi(i)}$. At the same time this means that $\sigma_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_{\pi(i)}) = 1 - \sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)})$ does not satisfy C and the clause remains signed $-$ after setting the value of $x_{\pi(i)}$. In both cases the variable $x_{\pi(i)}$ is deleted and the identity (1) still holds. On the other hand if $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)})$ does not satisfy C when $\mathbf{u}$ is used, then (since it was signed $+$) the clause C is now satisfied and disappears from the formula. But at the same time this means $\sigma_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_{\pi(i)})$ satisfies C, since it was signed $-$ under $\bar{\mathbf{u}}$, and therefore C is satisfied again and disappears from the formula. The variable $x_{\pi(i)}$ is deleted in both cases and again (1) is verified.

The case when clause C is signed $-$ under $\mathbf{u}$ and signed $+$ under $\bar{\mathbf{u}}$ is considered similarly. Finally, suppose $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)})$ violates a clause C containing $x_{\pi(i)}$. This means that C contains only this variable when setting this variable to $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)})$. By inductive assumption we see that the same is true under $\bar{\mathbf{u}}$. In both cases both the variable and clause are removed from the formula. This completes the proof of the inductive step.

Finally, since the distribution of $\mathbf{U}$ and $\bar{\mathbf{U}}$ is identical for i.i.d. sequences chosen uniformly at random from $[0, 1]$, we obtain (3). □

3.3 Influence ranges

We now define the notion of influence (which depends on the formula $\mathbf{\Phi}(n, dn)$ and ordering $\mathbf{Z}$, but not on further random choices of the τ-decimation algorithm). We introduce the following relationship between the variables $x_1, \ldots, x_n$ of our formula.

Definition 3.2. *Given a random formula $\mathbf{\Phi}(n, dn)$ and random sequence $\mathbf{Z}$ we say that x_i* influences *x_j if either $x_j = x_i$ or in the underlying node-to-node graph $\mathbb{G} = \mathbb{G}(\mathbf{\Phi}(n, dn))$ there exists a sequence of nodes $y_0, y_1, \ldots, y_t \in \{x_1, \ldots, x_n\}$ with the following properties:*

(i) $y_0 = x_i$ and $y_t = x_j$.

(ii) y_l and y_{l+1} are connected by a path of length at most r in graph $\mathbb{G}$ for all $l = 0, 1, \ldots, t-1$.

(iii) $Z_{y_{l-1}} > Z_{y_l}$ for $l = 0, 1, \ldots, t$. In particular, $Z_{x_i} > Z_{x_j}$.

In this case we write $x_i \rightsquigarrow x_j$. We denote by $\mathcal{IR}_{x_i}$ the set of variables x_j influenced by x_i and call it influence range *of x_i.*

Note that indeed the randomness underlying the sets $\mathcal{IR}_{x_i}, 1 \le i \le n$ as well as the relationship $\rightsquigarrow$ is the function of the randomness of the formula $\mathbf{\Phi}(n, dn)$ and vector $\mathbf{Z}$, but is independent from the random vector $\mathbf{U}$.

While the definition above is sound for every constant $r > 0$, we will apply it in the case where r is the parameter appearing in the context of τ-decimation algorithm. Namely, the τ function is applied to the set of rooted instances $\mathcal{SAT}_r$ defined above. In this case the notion of influence range is justified by the following observation.

Proposition 3.3. *Given realizations Φ and $\mathbf{z}$ of the random formula $\mathbf{\Phi}(n, dn)$ and random ordering $\mathbf{Z}$, suppose $\mathbf{u} = (u_i, 1 \le i \le n)$ and $\mathbf{u}' = (u_i', 1 \le i \le n)$ are such that $u_{i_0} = u_{i_0}'$ and $u_i = u_i'$ for all $i \ne i_0$. Then $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x) = \sigma_{\Phi,\mathbf{z},\mathbf{u}'}(x)$ for every $x \notin \mathcal{IR}_{i_0}$. That is, changing the values of $\mathbf{u}$ at i_0 may impact the decisions associated with only variables x in $\mathcal{IR}_{x_{i_0}}$*

Proof. Given a variable $x_i, i \ne i_0$, in order for its decision $\sigma_{\Phi,\mathbf{z},\cdot}(x_i)$ to be affected by switching from $\mathbf{u}$ to $\mathbf{u}'$, there must exist a variable x_{i_1} with distance at most r (with respect to the node-to-node graph $\mathbb{G} = \mathbb{G}(\Phi)$) from x_i such that $z_{x_{i_1}} > z_{x_i}$ and such that the decision for x_{i_1} is affected by the switch. In its turn such a variable exist if either $i_1 = i_0, z_{i_1} > z_{i_0}$ and $x_{i_0} \in B(x_i, r)$, and in particular $x_{i_0} \rightsquigarrow x_i$, or if there exists $x_{i_2} \in B(x_{i_1}, r)$ such that $z_{i_2} > z_{i_1}$ and x_{i_2} is affected by the switch. In this case again $x_{i_2} \rightsquigarrow x_i$. Continuing, we see that in order for x_i to be affected by the switch, it must be the case that $x_{i_0} \rightsquigarrow x_i$. □

We now obtain a probabilistic bound on the size of a largest in cardinality of the influence range classes $\mathcal{IR}_{x_i}, 1 \le i \le n$.

Proposition 3.4. *The following holds*

$$\lim_{n\to\infty} \mathbb{P}(\max_{1\le i\le n} |\mathcal{IR}_{x_i}| \ge n^{\frac{1}{3}}) = 0.$$

We defer the proof of this proposition to the appendix — see Section B. The choice of exponent $1/3$ is somewhat arbitrary here. In fact the bound holds for any exponent in $(0,1)$, and for our purposes, as we are about to see in Section 5, any constant in the range $(0,1/2)$ suffices.

4 The clustering property of random NAE-K-SAT problem

In this section we establish the clustering property of random NAE-K-SAT problem when d is large enough (in terms of K). Recall that the random NAE-K-SAT formula $\mathbf{\Phi}(n,dn)$ is satisfiable with probability approaching unity as $n \to \infty$, when $d \le d_s$, where $d_s = 2^{K-1}\ln 2 - \ln 2/2 - 1/4 - f(K)$ for some function $f(K)$ satisfying $\lim_{K\to\infty} f(K) = 0$. Recalling our notation for the set of satisfying assignment $\mathbb{SAT}(\Phi)$ of a formula Φ, we have $\mathbb{P}(\mathbb{SAT}(\mathbf{\Phi}(n,dn)) \ne \emptyset) \to 1$ as $n \to \infty$ for every $d < d_s$.

Let $\rho(\sigma^1,\sigma^2)$ denote the Hamming distance between two assignments σ^1 and σ^2. Namely, $\rho(\sigma^1,\sigma^2)$ is the number of variables x_i with different assignments according to σ^1 and σ^2.

We now state the main result of this section.

Theorem 4.1. *For al sufficiently large K and for all $d > (1/2)d_s$ there exists $0 < \delta_1(d) < \delta_2(d) < 1/2$ such that*

$$\lim_{n\to\infty} \mathbb{P}\left(\forall\ \sigma^1,\sigma^2 \in \mathbb{SAT}(\mathbf{\Phi}(n,dn)), \rho(\sigma^1,\sigma^2) \notin [\delta_1(d)n, \delta_2(d)n]\right) = 1.$$

In fact $\delta_1(d)$ and $\delta_2(d)$ can be chosen so that $\delta_1(d) \to 0$ and $\delta_2(d) \to 1/2$ as $d \to d_s$.

Namely, for all sufficiently large K and all $d > (1/2)d_s$, every two satisfying assignment either agree on at least $1 - \delta_1(d)$ fraction of variables or on at most $1 - \delta_2(d)$ fraction of variables.

Proof. The proof follows by a simple counting argument.

Fix $\alpha \in (\delta_1(d), \delta_2(d))$ so that αn is an integer, and let $\sigma^1, \sigma^2 \in \{0,1\}^n$ be at Hamming distance αn. We estimate the probability that $\Phi(n,dn)$ is satisfied by both σ^1 and σ^2. Denoting by $\mathbb{SAT}(C)$ the set of assignments which satisfy clause C, for a random clause of length K, the probability that C is satisfied by both σ^1 and σ^2 equals

$$\begin{aligned} &1 - 2\mathbb{P}(\sigma^1 \notin \mathbb{SAT}(C)) + \mathbb{P}(\sigma_1, \sigma^2 \notin \mathbb{SAT}(C)) \\ &= \ 1 - 2^{-(K-2)} + 2^{-(K-1)}(\alpha^K + (1-\alpha)^K). \end{aligned}$$

It follows that

$$\mathbb{P}\left(\sigma^1,\sigma^2 \in \mathbb{SAT}(\mathbf{\Phi}(n,dn))\right) = \Big(1 - 2^{-(K-2)} + 2^{-(K-1)}(\alpha^K + (1-\alpha)^K)\Big)^{dn}$$

and so

$$\mathbb{P}\left(\exists \sigma^1, \sigma^2 \in \mathbb{SAT}(\mathbf{\Phi}(n, dn)) \text{ s.t. } \rho(\sigma^1, \sigma^2) = \alpha n\right)$$
$$\leq 2^n 2^{H(\alpha)n} \left(1 - 2^{-(K-2)} + 2^{-(K-1)}(\alpha^K + (1-\alpha)^K)\right)^{dn},$$

where $H(\alpha)$ is the usual entropy binary function $-\alpha \ln \alpha - (1-\alpha)\ln(1-\alpha)$. The lemma now follows from simplifying the above expression. We note that for $0 < \alpha < 1$, as $K \to \infty$ the expression $1 - 2^{-(K-2)} + 2^{-(K-1)}(\alpha^K + (1-\alpha)^K) \approx 1 - 2^{-(K-2)} \approx e^{-2^{-(K-2)}}$. The value of d_s is such that $e^{-d_s 2^{-(K-1)}} \approx \frac{1}{2}$. Writing d as βd_s, we have $e^{-d2^{-(K-2)}} \leq 4^{-\beta}$ for all sufficiently large K. We conclude that

$$2^{(1+H(\alpha))n} \left(1 - 2^{-(K-2)} + 2^{-(K-1)}(\alpha^K + (1-\alpha)^K)\right)^{dn} \leq (2^{1+H(\alpha)-2\beta})^n = \exp(-\Omega(n)),$$

provided $H(\alpha) \leq 2\beta - 1$ or $\alpha < H^{-1}(2\beta - 1)$. Such non-trivial α exist provided $\beta > 1/2$ as assumed. The theorem thus holds for some $0 < \delta_1(d) < \delta_2(d) < H^{-1}(2\beta - 1)$. We note that as the ratio of $d/d_s \to 1$, we can let $\delta_2(d) \to \frac{1}{2}$, as asserted. □

5 Proof of Theorem 2.4

The main lemma of this section states that if a τ-decimation algorithm works well on random instances of NAE-K-SAT, then it can be run twice to produce two solutions of arbitrary amount of overlap, in particular in amounts forbidden by Theorem 4.1. We state our lemma below and show how Theorem 2.4 follows immediately. The rest of this section is devoted to the proof of the lemma.

We first recall some notation from Section 3. Given a local rule $\tau : \mathcal{SAT}_r \to [0,1]$, let $\sigma_{\Phi,\mathbf{Z},\mathbf{U}}$ denote the assignment produced by the τ-decimation algorithm on input Φ and ordering given by $\mathbf{Z}$ and using $\mathbf{U}$ to determine the rounding of the probabilities given by τ. Recall that $\mathbb{SAT}(\Phi)$ denotes the set of satisfying assignments of Φ. Recall that $\rho(\sigma^1, \sigma^2)$ denotes the Hamming distance between assigments σ^1 and σ^2. Let α_n denote the probability that τ-decimation algorithm finds a satisfying assignment in a random formula $\mathbf{\Phi}(n, dn)$. Namely, $\alpha_n = \mathbb{P}(\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{U}} \in \mathbb{SAT}(\mathbf{\Phi}(n, dn)))$.

Lemma 5.1. *Fix $r < \infty$ and let $\tau : \mathcal{SAT}_r \to [0,1]$ be a local rule and let $0 < \delta_1 < \delta_2 < 1/2$ be arbitrary. Suppose $\limsup_n \alpha_n > 0$. Then there exists $\beta > 0$ such that for every sufficiently large n:*

$$\liminf_n \mathbb{P}_{\mathbf{\Phi}(n,dn),\mathbf{Z}} \begin{pmatrix} \exists \mathbf{u}, \mathbf{v} \textit{ s.t. } \rho(\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{u}}, \sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{v}}) \in [\delta_1 n, \delta_2 n]) \\ \textit{and } \sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{u}}, \sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{v}} \in \mathbb{SAT}(\mathbf{\Phi}(n, dn)) \end{pmatrix} \geq \beta.$$

We defer the proof of Lemma 5.1 to the rest of this section. First we show how Theorem 2.4 follows immediately.

Proof of Theorem 2.4. We prove the result by contradiction. Our goal is to show that $\alpha_n \to 0$ as $n \to \infty$. Assume otherwise. Then the assumption of the lemma holds. Let δ_1, δ_2 be as given by Theorem 4.1. Then, by Lemma 5.1 we have with probability at least $\beta - o(1)$, $\Phi(n, dn)$ is such that that there exist $\sigma^1 = \sigma_{\Phi(n,dn),\mathbf{Z},\mathbf{u}}$ and $\sigma^2 = \sigma_{\Phi(n,dn),\mathbf{Z},\mathbf{u}}$ such that $\sigma^1, \sigma^2 \in \mathbb{SAT}(\Phi(n, dn))$ and $\rho(\sigma^1, \sigma^2) \in [\delta_1 n, \delta_2 n]$. This immediately contradicts Theorem 4.1. □

5.1 Proof of Lemma 5.1

We remark that the proof below would be straightforward if we could assume $\alpha_n = 1$ i.e, the τ-decimation algorithm always produces satisfying assigments. In this case we would pick two independent random sequences $\mathbf{U}^1$ and $\mathbf{U}^2$ and consider the assignments $\sigma^i = \sigma_{\Phi(n,dn),\mathbf{Z},\mathbf{U}^i}$ for $i \in \{1,2\}$. We easily have that $\rho(\sigma^1,\sigma^2) \approx n/2$. On the other hand we can change $\mathbf{U}^1$ to $\mathbf{U}^2$ one coordinate at a time, and the absence of long range effects (Proposition 3.4) implies each step produces a satisfying assigment with the Hamming distance changing only by $o(n)$ in each step. We conclude that somewhere along the way the overlap with the initial assigment σ^1 must be in the forbidden regime.

When $\alpha_n \neq 1$, the argument gets a bit more entangled, since we don't have that at each step the assignments produced are satisfying assignments. However we are able to show that the overlap at any particular stage is concentrated and while the probability that we find satisfying assigments remains positive. Details below.

Proof of Lemma 5.1. Given a random formula $\mathbf{\Phi}(n,dn)$ and a random sequence $\mathbf{Z}$ generating the order of setting the variables, let us consider two independent vectors $\mathbf{U}^1,\mathbf{U}^2$ which can be used to generate assignments. By definition we have

$$\mathbb{P}(\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{U}^1} \in \mathbb{SAT}(\mathbf{\Phi}(n,dn))) = \mathbb{P}(\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{U}^2} \in \mathbb{SAT}(\mathbf{\Phi}(n,dn))) = \alpha_n.$$

We now consider a sequence of vectors $\mathbf{V}^t, t=0,1,\ldots,n$ which interpolate between $\mathbf{U}^1$ and $\mathbf{U}^2$. Specifically, let $\mathbf{V}^t = (V_1^t,\ldots,V_n^t)$ where $V_i^t = U_i^2, i \le t$ and $V_i^t = U_i^1, t < i \le n$. Note that for every $t = 0,1,\ldots,n$, $\mathbf{V}$ is a vector of i.i.d. random variables with the uniform in $[0,1]$ distribution. Furthermore, $\mathbf{V}^0 = \mathbf{U}^1$ and $\mathbf{V}^n = \mathbf{U}^2$. Recall the notation $\mathcal{IR}_{x_t}$ for the influence region of variable x_t, i.e., all variables whose decision is potentially influenced by be assigment of x_t by the τ-decimation algorithm. Observe that given realizations $\mathbf{u}^1$ and $\mathbf{u}^2$ and the resulting realizations $\mathbf{v}^t$ of $\mathbf{V}^t$ we have

$$\rho(\sigma_{\Phi,\mathbf{z},\mathbf{v}^{t+1}},\sigma_{\Phi,\mathbf{z},\mathbf{v}^t}) \le |\mathcal{IR}_{x_{t+1}}|, \qquad 0 \le t \le n-1, \tag{4}$$

since, by Proposition 3.3 changing the value of u_{t+1} impacts only the decisions for variables in $\mathcal{IR}_{x_{t+1}}$. We now consider a realization Φ of a formula $\mathbf{\Phi}(n,dn)$ and realization $\mathbf{z}$ of the order $\mathbf{Z}$ and the corresponding sequence of sets $\mathcal{IR}_{x_i}, 1 \le i \le n$, which are uniquely determined by Φ and $\mathbf{z}$. Let $\mathcal{E}_n$ denote the event (the set of Φ and $\mathbf{z}$) that $\max_{1\le i\le n} |\mathcal{IR}_{x_i}| \le n^{1/3}$. In particular, by Proposition 3.4 we have

$$\lim_{n\to\infty} \mathbb{P}(\mathcal{E}_n) = 1. \tag{5}$$

Fix arbitrary $0 < \delta_1 < \delta_2 < 1/2$ as in Theorem 4.1. Let n be large enough so that $n^{1/3} < (\delta_2 - \delta_1)n$. We have by Lemma 3.1 that for every Φ and $\mathbf{z}$,

$$\mathbb{E}[\rho(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1},\sigma_{\Phi,\mathbf{z},\mathbf{U}^2})] = n/2.$$

Then if Φ and $\mathbf{z}$ are realizations such that the event $\mathcal{E}_n$ takes place, then we can find $t_0 = t_0(\Phi,\mathbf{z})$ such that

$$\mathbb{E}[\rho(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1},\sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0}})] \in \left[\frac{\delta_1+\delta_2}{2}n - n^{1/3}, \frac{\delta_1+\delta_2}{2}n + n^{1/3}\right],$$

since by (4) the expected increments $E[\rho(\sigma_{\Phi,\mathbf{z},\mathbf{v}^{t+1}},\sigma_{\Phi,\mathbf{z},\mathbf{v}^t})]$ are bounded by $n^{1/3}$. We now argue that in fact $\rho(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1},\sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0}})$ is concentrated around its mean as $n \to \infty$. The distance is a function of $n+t_0$ i.i.d. random variables $U_1^1,\ldots,U_n^1,U_1^2,\ldots,U_{t_0}^2$. Further, changing any one of these $n+t_0$ random variables changes the distance ρ by at most $n^{1/3}$ again by Proposition 3.3. Applying Azuma's inequality

$$\begin{aligned}
&\mathbb{P}\left(\left|\rho(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1},\sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0}}) - \frac{\delta_1+\delta_2}{2}n\right| \ge \frac{\delta_2-\delta_1}{4}n\right)\\
&\le 2\exp\left(-\frac{(\frac{\delta_2-\delta_1}{4}n - n^{\frac{1}{3}})^2}{2(n+t_0)n^{\frac{2}{3}}}\right)\\
&= \exp(-\delta_3 n^{1/3}),
\end{aligned}$$

for appropriately small $\delta_3 > 0$, and the concentration is established. The event

$$\left|\rho(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0}}) - \frac{\delta_1 + \delta_2}{2} n\right| < \frac{\delta_2 - \delta_1}{4} n$$

implies the event

$$\rho(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0}}) \in [\delta_1 n, \delta_2 n].$$

We conclude that for every Φ and $\mathbf{z}$ such that the event $\mathcal{E}_n$ takes place, we have

$$\lim_n \mathbb{P}\left(\rho(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0}}) \in [\delta_1 n, \delta_2 n]\right) = 1. \tag{6}$$

For completion, let us set $t_0 = 0$ when Φ and $\mathbf{z}$ are such that the event $\mathcal{E}_n$ does not take place. Letting $T = t_0(\mathbf{\Phi}(n, dn), \mathbf{Z})$ to be thus defined random variable, observe that $\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{V}^T}$ has the same distribution as $\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{U}^1}$ since the random variable T only affects the indices i for which we are using variables U_i^1 vs U_i^2, and since vectors $\mathbf{U}^1, \mathbf{U}^2$ are independent from $\mathbf{\Phi}(n, dn)$ and $\mathbf{Z}$. Therefore, $\mathbb{P}(\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{V}^T} \in \mathbb{SAT}(\mathbf{\Phi}(n, dn))) = \alpha_n$. Furthermore, for every $\Phi, \mathbf{z}$ such that the event $\mathcal{E}_n$ takes place and the corresponding t_0, since $U_i^1, i \le t$ and $U_i^2, i \le t$ are independent, then

$$\begin{aligned}
&\mathbb{P}_{\mathbf{U}^1,\mathbf{U}^2}(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0}} \in \mathbb{SAT}(\Phi)) \\
&\quad = \quad \mathbb{E}_{\mathbf{U}^1,\mathbf{U}^2}[\mathbf{1}(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0}} \in \mathbb{SAT}(\Phi))] \\
&\quad = \quad \mathbb{E}_{U^1_{t_0+1},\ldots,U^1_n}[\mathbb{E}_{U^1_1,\ldots,U^1_{t_0},U^2_1,\ldots,U^2_{t_0}}[\mathbf{1}(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0}} \in \mathbb{SAT}(\Phi)) \mid U^1_{t_0+1}, \ldots, U^1_n]] \\
&\quad = \quad \mathbb{E}_{U^1_{t_0+1},\ldots,U^1_n}[\mathbb{E}^2_{U^1_1,\ldots,U^1_{t_0}}[\mathbf{1}(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1} \in \mathbb{SAT}(\Phi)) \mid U^1_{t_0+1}, \ldots, U^1_n]] \\
&\quad \ge \quad \mathbb{E}^2_{U^1_{t_0+1},\ldots,U^1_n}[\mathbb{E}_{U^1_1,\ldots,U^1_{t_0}}[\mathbf{1}(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1} \in \mathbb{SAT}(\Phi)) \mid U^1_{t_0+1}, \ldots, U^1_n]] \\
&\quad = \quad \mathbb{E}^2_{\mathbf{U}^1}[\mathbf{1}(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1} \in \mathbb{SAT}(\Phi)) \ ] \\
&\quad = \quad \mathbb{P}^2_{\mathbf{U}^1}(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1} \in \mathbb{SAT}(\Phi)).
\end{aligned}$$

On the other hand, if Φ and $\mathbf{z}$ are such that the event $\mathcal{E}_n$ does not take place then $\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{V}^T} = \sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{U}^1}$. Thus

$$\mathbb{P}(\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{U}^1}, \sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{V}^T} \in \mathbb{SAT}(\mathbf{\Phi}(n, dn))) \ge \alpha_n^2.$$

We finally put together our observations together as follows:

$$\begin{aligned}
\alpha_n^2 \quad &\le \quad \mathbb{P}(\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{U}^1}, \sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{V}^T} \in \mathbb{SAT}(\mathbf{\Phi}(n, dn))) \\
&\le \quad \mathbb{P}(\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{U}^1}, \sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{V}^T} \in \mathbb{SAT}(\mathbf{\Phi}(n, dn)), \mathcal{E}_n, \rho(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^T}) \in [\delta_1 n, \delta_2 n]) \\
&\quad\quad + \mathbb{P}(\mathcal{E}_n, \rho(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^T}) \notin [\delta_1 n, \delta_2 n]) \\
&\quad\quad + \mathbb{P}(\mathcal{E}_n^c) \\
&\le \quad \mathbb{P}(\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{U}^1}, \sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{V}^T} \in \mathbb{SAT}(\mathbf{\Phi}(n, dn)), \rho(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^T}) \in [\delta_1 n, \delta_2 n]) \\
&\quad\quad + \mathbb{P}(\mathcal{E}_n, \rho(\sigma_{\Phi,\mathbf{z},\mathbf{U}^1}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^T}) \notin [\delta_1 n, \delta_2 n]) \\
&\quad\quad + \mathbb{P}(\mathcal{E}_n^c)
\end{aligned}$$

Now by (6) we have that the second probability converges to zero as well. Finally, we have (5). We obtain the lemma for any $0 < \beta < \limsup_n \alpha^2$ and $\mathbf{U} = \mathbf{U}^1$ and $\mathbf{V} = \mathbf{V}^T$.

□

References

[BMZ05] A. Braunstein, M. Mézard, and R. Zecchina, *Survey propagation: An algorithm for satisfiability*, Random Structures & Algorithms **27** (2005), no. 2, 201–226.

[CO11] A. Coja-Oghlan, *On belief propagation guided decimation for random k-sat*, Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2011, pp. 957–966.

[COP12] A. Coja-Oglan and K. Panagiotou, *Catching the k-naesat threshold*, Proceedings of the 44th symposium on Theory of Computing, ACM, 2012, pp. 899–908.

[Fri99] E. Friedgut, *Sharp thresholds of graph proprties, and the k-SAT problem*, J. Amer. Math. Soc. **4** (1999), 1017–1054.

[GG10] D. Gamarnik and D. Goldberg, *Randomized greedy algorithms for independent sets and matchings in regular graphs: Exact results and finite girth corrections.*, Combinatorics, Probability and Computing **19** (2010), 61–85.

[GS13] D. Gamarnik and M. Sudan, *Limits of local algorithms over sparse random graphs*, arXiv preprint arXiv:1304.1831 (2013).

[HLS] H. Hatami, L. Lovász, and B. Szegedy, *Limits of local-global convergent graph sequences*, Preprint at http://arxiv.org/abs/1205.4356.

[KMRT$^+$07] F. Krzakała, A. Montanari, F. Ricci-Tersenghi, G. Semerjian, and L. Zdeborová, *Gibbs states and the set of solutions of random constraint satisfaction problems*, Proceedings of the National Academy of Sciences **104** (2007), no. 25, 10318–10323.

[MM] M. Mezard and A. Montanari, *Reconstruction on trees and spin glass transition*, J. Stat. Phys., to appear.

[MM09] ______, *Information, physics and computation*, Oxford graduate texts, 2009.

[MMW07] Elitza Maneva, Elchanan Mossel, and Martin J Wainwright, *A new look at survey propagation and its generalizations*, Journal of the ACM (JACM) **54** (2007), no. 4, 17.

[MPZ02] Marc Mézard, Giorgio Parisi, and Riccardo Zecchina, *Analytic and algorithmic solution of random satisfiability problems*, Science **297** (2002), no. 5582, 812–815.

[NO08] Huy N. Nguyen and Krzysztof Onak, *Constant-time approximation algorithms via local improvements*, FOCS, IEEE Computer Society, 2008, pp. 327–336.

A Survey Propagation for random NAE-K-SAT

We now turn to the Survey Propagation algorithm and SP-guided decimation algorithm. The setup is similar to the one for BP. In particular in steps $i = 1, 2, \ldots, n$ certain marginal value is computed and the decision for x_i is again based on this marginal value, except now the marginal values do not correspond to the ratio of the number of assignments, but rather correspond to ratios when the problem is lifted to a new certain constraint satisfaction problem with decision variables $0, 1, *$. We do not describe here the rationale for this lifting procedure, as this has been documented in many papers, including [BMZ05],[MMW07],[MPZ02],[MM09]. Instead we present the SP algorithm and SP-guided decimation algorithm formally, following closely [MM09] with the appropriate adjustment from the K-SAT problem to the NAE-K-SAT problem, and then convince ourselves that SP-guided decimation algorithm is again the special case of a τ-decimation algorithm.

We will then be able to conclude that SP-guided decimation algorithm fails to find a satisfying assignment with probability approaching unity, in the regime outlined in our main result, Theorem 2.4.

The SP algorithm is an iterative scheme described as follows. The details and notations are very similar to the ones described in [MM09]. Specifically iterations (7)-(11) correspond to iterations (20.17)-(20.20) in this book. Consider an arbitrary reduced or non-reduced NAE-K-SAT formula Φ on variables $x_1, \ldots, x_N$. For each iteration $t = 0, 1, \ldots$, each variable/clause pair (x, C) such that x appears in C (namely there is an edge between x and C in the bi-partite factor graph representation) is associated with five random variables $Q^t_{x,C,U}, Q^t_{x,C,S}, Q^t_{x,C,*}, \hat{Q}^t_{C,x,S}$ and $\hat{Q}^t_{C,x,U}$. Here is the interpretation of these variables. Each of them is a message send from variable to a clause containing this variable or a message from a clause to a variable which belongs to this clause. Specifically, $Q^t_{x,C,U}(Q^t_{x,C,S})$ is interpreted as the probability computed at iteration t that the variable x is forced by clauses D other than C to take value which does not (does) satisfy U. $Q^t_{x,C,*}$ is interpreted that none of these forcing takes place. $\hat{Q}^t_{C,x,S}$ is interpreted as probability computed at iteration t that all variables $y \in C$ other than x do not satisfy C, and thus the only hope of satisfying C is for x to do so. Similarly, $\hat{Q}^t_{C,x,S}$ is the probability that all variables y in C other than x do satisfy C and thus the only hope of satisfying clause C is for x to violate it. The latter case is an artifact of the NAE variant of the problem and need not be introduced in the SP iterations for the K-SAT problem.

The variables Q^t and $\hat{Q}^t$ are then computed as follows. At time $t = 0$ the variables are generated uniformly at random from $[0, 1]$ independently for all five variables. Then they are normalized so that $Q^0_{x,C,U} + Q^0_{x,C,S} + Q^0_{x,C,*} = 1$, which is achieved by dividing each term by the sum $Q^0_{x,C,U} + Q^0_{x,C,S} + Q^0_{x,C,*}$. The variables $\hat{Q}^0_{C,x,S}$ and $\hat{Q}^0_{C,x,U}$ do not need to be normalized.

Now we describe the iteration procedures at times $t \geq 0$. For each such pair x, C, let $\mathcal{S}_{x,C}$ be the set of clauses containing x other than C, in which x appears in the same way as in C. Namely if x appears in C without negation, it appears without negation in clauses in $\mathcal{S}_{x,C}$ as well. Similarly, if x appears as $\bar{x}$ in C, the same is true for clauses in $\mathcal{S}_{x,C}$. Let $\mathcal{U}_{x,C}$ be the remaining set of clauses containing x, namely clauses, where x appears opposite to the way it appears in C. Now for each $t = 0, 1, 2, \ldots$ assume $Q^t_{x,C,U}, Q^t_{x,C,S}, Q^t_{x,C,*}, \hat{Q}^t_{C,x,S}$ and $\hat{Q}^t_{C,x,U}$ are defined. Define the random variable $\hat{Q}^{t+1}_{x,C,S}$ and $\hat{Q}^{t+1}_{x,C,U}$ as follows. Suppose C is unsigned in Φ. Then

$$\hat{Q}^{t+1}_{C,x,S} = \prod_{y \in C \setminus x} Q^t_{y,C,U}, \tag{7}$$

and

$$\hat{Q}^{t+1}_{C,x,U} = \prod_{y \in C \setminus x} Q^t_{x,C,S}. \tag{8}$$

Here $C \setminus x$ is the set of variables in clause C other than x. The interpretation for this identities is as follows. When C is not signed, the clause C forces its variable x to satisfy it if all other variables y in C where forced not to satisfy C at previous iteration due to other clauses. The first identity is the probability of this event assuming the events "y is forced not to satisfy C" are independent. The second identity is interpreted similarly, though it is only relevant only for NAE-K-SAT problem and does not appear for the corresponding iterations for the K-SAT problem.

If the clause C is signed $+$, then we set $\hat{Q}^{t+1}_{C,x,S} = 0$ and

$$\hat{Q}^{t+1}_{C,x,U} = \prod_{y \in C \setminus x} Q^t_{x,C,S}. \tag{9}$$

The interpretation is that if C is signed $+$, then one of the variables was already set to satisfy it. Thus the only way the clause C can force x not to satisfy it is if all other variables y are forced to satisfy C. Again this is only relevant for the NAE-K-SAT problem. Similarly, if C is signed $-$, then $\hat{Q}^{t+1}_{C,x,U} = 0$ and

$$\hat{Q}^{t+1}_{C,x,S} = \prod_{y \in C \setminus x} Q^t_{x,C,U}. \tag{10}$$

Next we define variables $R^{t+1}_{x,C,S}, R^{t+1}_{x,C,U}$ and $R^{t+1}_{x,C,*}$ which stand for $Q^{t+1}_{x,C,S}, Q^{t+1}_{x,C,U}$ and $Q^{t+1}_{x,C,*}$ before the normalization. These random variables are computed using the following rules:

$$R^{t+1}_{x,C,S} = \prod_{D \in \mathcal{U}_{x,C}} (1 - \hat{Q}^t_{D,x,S}) \left(1 - \prod_{D \in \mathcal{S}_{x,C}} (1 - \hat{Q}^t_{D,x,S})\right) \tag{11}$$

$$+ \prod_{D \in \mathcal{S}_{x,C}} (1 - \hat{Q}^t_{D,x,U}) \left(1 - \prod_{D \in \mathcal{U}_{x,C}} (1 - \hat{Q}^t_{D,x,U})\right), \tag{12}$$

which is interpreted as follows. Variable x is forced to satisfy clause C at iteration $t+1$, if at iteration t, either at least one of the clauses D containing x in the same way as C (namely one of the clauses in $\mathcal{S}_{x,C}$) forces x to take value which satisfies C, and none of the clauses in $\mathcal{U}_{x,C}$ force x to take value which violates C (as otherwise a contradiction would be reached), or alternatively, if at iteration t, at least one of the clauses D containing x in the way opposite to C (namely one of the clauses in $\mathcal{U}_{x,C}$) forces x to take value which satisfies C, and none of the clauses in $\mathcal{S}_{x,C}$ force x take value which violates C (since otherwise again the contradiction would be reached). Variable $R^{t+1}_{x,C,S}$ represents the probability of this forcing which is expressed in probabilities of the corresponding forcing events at time t, again assuming independence.

Similarly, define

$$R^{t+1}_{x,C,U} = \prod_{D \in \mathcal{S}_{x,C}} (1 - \hat{Q}^t_{D,x,S}) \left(1 - \prod_{D \in \mathcal{U}_{x,C}} (1 - \hat{Q}^t_{D,x,S})\right)$$

$$+ \prod_{D \in \mathcal{U}_{x,C}} (1 - \hat{Q}^t_{D,x,U}) \left(1 - \prod_{D \in \mathcal{S}_{x,C}} (1 - \hat{Q}^t_{D,x,U})\right).$$

The interpretation for $R^{t+1}_{x,C,U}$ is similar. Next, define

$$R^{t+1}_{x,C,*} = \prod_{D \in \mathcal{S}_{x,C} \cup \mathcal{U}_{x,C}} (1 - \hat{Q}^t_{D,x,S})(1 - \hat{Q}^t_{D,x,U}).$$

$R^{t+1}_{x,C,*}$ is interpreted as the probability that x is not forced in either way by constraints other than C. Finally, we let $U^{t+1}_{x,C,S}, U^{t+1}_{x,C,U}$ and $U^{t+1}_{x,C,*}$ to be quantities $R^{t+1}_{x,C,S}, R^{t+1}_{x,C,U}$ and $R^{t+1}_{x,C,*}$, respectively, normalized by their sum $R^{t+1}_{x,C,S} + R^{t+1}_{x,C,U} + R^{t+1}_{x,C,*}$, so that the three variables sum up to one. The iterations (7)-(11) are conducted for some number of steps $t = 0, 1, \ldots, r$. Next variables $W_x(1)$ and $W_x(0)$ and $W_x(*)$ are computed for all variables x as follows. Let $\mathcal{S}_x$ be the set of clauses where x

appears without negation and let $\mathcal{U}_x$ be the set of clauses where x appears with negation. Then set

$$W_x(1) = \prod_{D\in\mathcal{U}_x}(1-\hat{Q}^r_{D,x,S})\left(1-\prod_{D\in\mathcal{S}_x}(1-\hat{Q}^r_{D,x,S})\right)$$
$$+\prod_{D\in\mathcal{S}_x}(1-\hat{Q}^r_{D,x,U})\left(1-\prod_{D\in\mathcal{U}_x}(1-\hat{Q}^r_{D,x,U})\right),$$

$W_x(1)$ is interpreted as probability (after normalization) that variable x is forced to take value 1, but is not forced to take value zero by all of the constraints containing x. Similarly, we have

$$W_x(0) = \prod_{D\in\mathcal{S}_x}(1-\hat{Q}^r_{D,x,S})\left(1-\prod_{D\in\mathcal{U}_x}(1-\hat{Q}^r_{D,x,S})\right)$$
$$+\prod_{D\in\mathcal{U}_x}(1-\hat{Q}^r_{D,x,U})\left(1-\prod_{D\in\mathcal{S}_x}(1-\hat{Q}^r_{D,x,U})\right),$$

with a similar interpretation. Then set

$$W_x(*) = \prod_{D\in\mathcal{S}_x}(1-\hat{Q}^r_{D,x,S}-\hat{Q}^r_{D,x,U})\prod_{D\in\mathcal{U}_x}(1-\hat{Q}^r_{D,x,S}-\hat{Q}^r_{D,x,U}),$$

which is interpreted as the probability (after normalization) that x is not take forced to be either 0 or 1. Finally, the values $W_x(0), W_x(1), W_x(*)$ are normalized to sum up to one. For simplicity we use the same notation for these quantities after normalization.

The random variables $W_x(0), W_x(1), W_x(*)$ are used to guide the decimation algorithm as follows. Given a random formula $\mathbf{\Phi}(n, dn)$, variable x_1 is selected. The random quantities $W_{x_1}(0), W_{x_1}(1)$ and $W_{x_1}(*)$ are computed and x_1 is set to 1 if $W_{x_1}(1) > W_{x_1}(0)$ and set it to zero otherwise. The formula is now reduced and contains variables $x_2, x_3, \ldots, x_n$. Variable x_2 is then selected and the random quantities $W_{x_2}(0), W_{x_2}(1)$ are computed with respect to the reduced formula. Then $W_{x_2}(*)$ and x_2 is set to 1 if $W_{x_2}(1) > W_{x_2}(0)$, and set it to zero otherwise. The procedure is repeated until all variables are set. This defines the SP-guided decimation algorithm.

It is clear again that the SP-guided decimation algorithm is the special case of τ-decimation algorithm, where τ function corresponds to the probability of the event $W_x(1) > W_x(0)$, when it applies to a reduced instance $B(x, r)$ with x as its root. The depth r of the instance corresponds to the number of iterations of the SP procedure. It is also clear that there is no inherent bias in the SP to set variable to 0 vs 1, the rule τ is balanced and thus Theorem 2.4 becomes applicable and Corollary 2.6 holds.

We note that, as for the BP case, the experiments based on the SP-guided decimation algorithm, instead choose variables with a largest bias $|W_x(1) - W_x(0)|$, among all of the remaining variables, as opposed to simply the next variable in the original order $x_1, \ldots, x_n$. But again no explanation was put forward saying that this biased version of the SP-guided decimation is critical for its success. Based on the statistical physics predictions, the non-size-biased version, namely the one presented above should also succeed in finding a satisfying assignment. Similarly, the number of iterations r was not fixed in the experiments. Instead the iterations were carried out until approximate convergence was achieved. Again, as in the case of BP-guided decimation algorithm, it appears that this was just a sensible implementation choice rather than a rule based on the statistical physics theory per se.

B Proof sketch of Proposition 3.4

Fix a variable x in $\mathbf{\Phi}(n, dn)$. We first establish an upper bound on the number of variables in a neighborhood $B(x, t)$ of x in the node-to-node graph $\mathbb{G}(\mathbf{\Phi}(n, dn))$ when t is moderately growing.

Lemma B.1. *There exists $\delta > 0$ and $\epsilon = \epsilon(\delta) < 1/3$ such that for all sufficiently large n*

$$\mathbb{P}(|B(x,t)| \geq n^{\epsilon}) \leq \frac{1}{n^2},$$

when $t \leq \delta \ln n$.

Proof. It is well known that for small enough $\delta > 0$ and $t = \delta \ln n$, the $B(x,t)$ is distributed approximately as a Poisson branching process with the off-spring distribution being Poisson with parameter $\beta \triangleq dK$. Furthermore, by increasing the number of clauses by $o(n)$ the Poisson branching process stochastically dominates the distribution of $B(x,t)$. Thus we obtain instead an upper bound on the number of off-springs in the t generations of a Poisson branching process with parameter β. Letting W_l denote the size of the l-th generation, our goal is then to obtain a bound on $\sum_{l \leq t} W_l$. We claim that for some $\epsilon = \epsilon(\delta)$ satisfying $\epsilon(\delta) \to 0$ as $\delta \to 0$, the following upper bound holds for each $W_l, l \leq t = \delta \ln n$:

$$\mathbb{P}(W_l > n^{\epsilon/2}) \leq \exp(-n^{\epsilon/4}), \tag{13}$$

from which the claim of the lemma follows by a union bound. To establish this bound we rely on the following known representation of the probability generating function of W_l. That is, let $G(\theta) = \mathbb{E}[\theta_1^W]$ for $\theta > 0$, where W_1 has Poisson mean β distribution. Then $G(\theta) = \exp(\beta\theta - \beta)$ and $\mathbb{E}[\theta^{W_l}] = G^{(l)}(\theta)$ - the l-th iterate of function $G(\theta)$. Now we let $\theta = 1 + \frac{1}{(e\beta)^t}$. Define $\gamma_l = 1/(e\beta)^l$. We now obtain an upper bound on $G^{(l)}(\theta)$. We have

$$G^{(1)}(\theta) = \exp(\beta\theta - \beta) = \exp(\beta\gamma_t) \leq 1 + \gamma_{t-1},$$

where we have used $\beta\gamma_t < 1$ and inequality $e^z \leq 1 + ez$ for $z \leq 1$. Then

$$G^{(2)}(\theta) = \exp(\beta G^{(1)}(\theta) - \beta) \leq \exp(\beta\gamma_{t-1}) \leq 1 + \gamma_{t-2},$$

since $\beta\gamma_{t-1} < 1$. Continuing, we obtain $G^{(l)}(\theta) \leq 1 + \gamma_{t-l}, 1 \leq l \leq t$. Applying this bound

$$\begin{aligned}
\mathbb{P}(Z_l \geq n^{\epsilon/2}) &= \mathbb{P}(\theta^{Z_l} \geq \theta^{n^{\epsilon/2}}) \\
&\leq \theta^{-n^{\epsilon/2}} \mathbb{E}[\theta^{Z_l}] \\
&\leq \theta^{-n^{\epsilon/2}} (1 + \gamma_{t-l}) \\
&\leq 2\theta^{-n^{\epsilon/2}}.
\end{aligned}$$

Now

$$\begin{aligned}
\theta^{-n^{\epsilon/2}} &= \exp(-n^{\epsilon/2} \ln(\theta)) \\
&= \exp(-n^{\epsilon/2}(\gamma_t + o(\gamma_t))).
\end{aligned}$$

Now since $t = \delta n$, then $\gamma_t = (e\beta)^t = n^{-\ln(e\beta)\delta}$, implying the upper bound $\exp(-n^{\epsilon/4})$ for large enough n when $\epsilon(\delta) > 2\ln(e\beta)\delta$. This completes the proof of the bound (13) and of lemma. □

Now we complete the proof of the Proposition. Applying union bound we have that that for every $\epsilon > 0$, $|B(x_i,t)| \leq n^{\epsilon}$) for all $i = 1, \ldots, n$ with probability approaching unity as $n \to \infty$. Given two variables x_i and x_j if $x_i \rightsquigarrow x_j$ and the distance in $\mathbb{G}(\mathbf{\Phi}(n, dn))$ between x_i and x_j is at least t, then there must exist $x_k \in B(x_i,t) \setminus B(x_i, t-1)$ such that $x_i \rightsquigarrow x_k$. Given a sequence $y_0 = x_i, y_1, \ldots, y_t = x_k$, with $x_k \in B(x_i,t) \setminus B(x_i,t-1)$, the probability of an event $Z_{y_l} > Z_{y_{l+1}}, 0 \leq l \leq t-1$ is $1/(t+1)!$. The total number of paths between x_i and variables in $B(x_i,t) \setminus B(x_i,t-1)$ is trivially at most $B(x_i,t)$, since $B(x_i,t)$ is tree. Thus, conditioned on $B(x_i,t)$, the expected number of variables in $B(x_i,t)$ is at most $B(x_i,t)r^t/(t+1)!$, where the extra factor r^t is due to choices of points $y_1, \ldots, y_t$ on a given path. When $t = \epsilon \ln$, the expected number is $B(x_i,t)n^{-\Omega(\ln \ln n)}$. Applying the bound Lemma B.1 and a union bound over x_i we obtain the result.