Performance of Sequential Local Algorithms for the Random NAE-K-SAT Problem

SIAM J. COMPUT.
Vol. 46, No. 2, pp. 590–619

PERFORMANCE OF SEQUENTIAL LOCAL ALGORITHMS FOR THE RANDOM NAE-K-SAT PROBLEM*

DAVID GAMARNIK† AND MADHU SUDAN‡

Abstract. We formalize the class of "sequential local algorithms" and show that these algorithms fail to find satisfying assignments on random instances of the "Not-All-Equal-K-SAT" (NAE-K-SAT) problem if the number of message passing iterations is bounded by a function moderately growing in the number of variables and if the clause-to-variable ratio is above $(1+o_K(1))\frac{2^{K-1}}{K}\ln^2 K$ for sufficiently large K. Sequential local algorithms are those that iteratively set variables based on some local information and/or local randomness and then recurse on the reduced instance. Our model captures some weak abstractions of natural algorithms such as Survey Propagation (SP)-guided as well as Belief Propagation (BP)-guided decimation algorithms—two widely studied message-passing–based algorithms—when the number of message-passing rounds in these algorithms is restricted to be growing only moderately with the number of variables. The approach underlying our paper is based on an intricate geometry of the solution space of a random NAE-K-SAT problem. We show that above the $(1+o_K(1))\frac{2^{K-1}}{K}\ln^2 K$ threshold, the overlap structure of m-tuples of nearly (in an appropriate sense) satisfying assignments exhibit a certain behavior expressed in the form of some constraints on pairwise distances between the m assignments for appropriately chosen positive integer m. We further show that if a sequential local algorithm succeeds in finding a satisfying assignment with probability bounded away from zero, then one can construct an m-tuple of solutions violating these constraints, thus leading to a contradiction. Along with [D. Gamarnik and M. Sudan, *Ann. Probab.*, to appear], where a similar approach was used in a (somewhat simpler) setting of nonsequential local algorithms, this result is the first work that directly links the overlap property of random constraint satisfaction problems to the computational hardness of finding satisfying assignments.

Key words. random graphs, algorithms, statistical physics

AMS subject classifications. 60C05, 82B20, 05C80

DOI. 10.1137/140989728

1. Introduction.

In this work we consider a class of algorithms which we dub "sequential local algorithms," that capture local implementations of message-passing–based decimation algorithms, including the Belief Propagation (BP)-guided and the Survey Propagation (SP)-guided decimation algorithms. We analyze the behavior of local sequential algorithms on random instances of "Not-All-Equal-K-SAT" (NAE-K-SAT). We describe the NAE-K-SAT problem and our class of algorithms, in that order, below. Later we explain how this class of algorithms is motivated by our attempt to understand our ability to analyze the performance of the message-passing–based decimation algorithms.

1.1. Our setting and results.

The NAE-K-SAT problem is a Boolean constraint satisfaction problem closely related to the more commonly studied K-SAT problem. An instance of the NAE-K-SAT problem consists of a collection of N K-clauses on n Boolean variables $x_1,\dots,x_n$. Each K-clause is given by K-literals, where each literal is either one of the variables or its negation. The clause is satisfied by a Boolean assignment to the variables if at least one of the literals is satisfied (set to

*Received by the editors October 1, 2014; accepted for publication (in revised form) November 22, 2016; published electronically March 7, 2017.
http://www.siam.org/journals/sicomp/46-2/98972.html
Funding: The first author's work was supported by NSF grant CMMI-1335155.
†Operations Research Center and Sloan School of Management, MIT, Cambridge, MA 02140 (gamarnik@mit.edu).
‡Microsoft Research New England, Cambridge, MA 02142 (madhu@mit.edu).

1) and at least one is unsatisfied (set to 0). (This symmetry between satisfied and unsatisfied literals lends a convenient symmetry to the NAE-K-SAT problem that is not shared by the K-SAT counterpart.) The collection of N clauses is satisfied by a Boolean assignment if each clause is satisfied. Given $\ell \leq N$, we say that a Boolean assignment ℓ-satisfies the collection if at most ℓ clauses are violated.

In this work we consider the ability to find satisfying and ℓ-satisfying assignments to random instances of the NAE-K-SAT problem. Here the N clauses are chosen uniformly and independently from the collection of $2^K \cdot \binom{n}{K}$ possible K-clauses. Throughout the paper we consider the regime where the number of variables n grows but the clause size K remains constant. In particular we consider the setting where $N = d \cdot n$ for some constant $d = d(K)$, that depends on K but not n, and consider what is the largest d for which there exists a efficient algorithm for identifying an ℓ-satisfying assignment with probability bounded away from zero as $n \to \infty$ by some function going to zero at some rate with n. We will be interested primarily in the regime where ℓ is a linear function of N and therefore n as well. The parameter d is often referred to as the *formula density*. Of course, no algorithm can find a satisfying assignment if none exists, and the limit of when such an assignment exists has been well studied. In particular Coja-Oghlan and and Panagiotou [COP12] have established that random instances of the NAE-K-SAT problem are satisfiable with high probability (w.h.p.) when the density d is below $d_s \triangleq 2^{K-1} \ln 2 - \ln 2/2 - 1/4 - o_K(1)$, and is not satisfiable w.h.p. when $d > d_s$. An earlier bound was obtained by Achlioptas and Moore [AM06] in the form $d_s = 2^{K-1} \ln 2 - O_K(1)$. Here $o_K(\cdot)$ and $O_K(\cdot)$ denote standard order-of-magnitude notation as K increases. Our interest is in determining how qualitatively close to this threshold an efficient algorithm can get; i.e., how does the largest density at which the algorithm manages to find a satisfying or even ℓ-satisfying assignment compare with d_s.

The class of algorithms that we explore in this work is what we call "sequential local algorithms." A sequential local algorithm can be described roughly as follows. The algorithm works by assigning Boolean values to variables sequentially, where a chosen variable is assigned its value by a potentially probabilistic choice, which depends on the local neighborhood of the variable at the time the choice is made. The local neighborhood is defined to be the graph-theoretic $B(r)$ ball of small value r radius with respect to the underlying factor graph on the set of variables and clauses, to be defined later. Once a variable is assigned a value, the formula is simplified (removing some clauses and restricting others). This in turn may influence the local neighborhoods of other variables, and when the future variables are set to particular Boolean values, this is done with respect to thus possibly modified neighborhoods. The algorithm continues with its iterations till all variables are set.

Local sequential algorithms capture restricted versions of BP- and SP-guided decimation algorithms, specifically when the number of message passing iterations used between every decimation step is bounded by $O(r)$. (BP- and SP-guided decimation algorithms really form a very general class with many possible implementations and interpretations. In section 1.3 we discuss the specific assumptions we make and their potential limitations.) In the specific context of a BP-guided decimation algorithm based on r iterations, the local rule assigns value 1 to a variable x with probability equal to the fraction of assignments in which x is assigned value 1 among all assignments that satisfy all clauses in the local neighborhood $B(r)$. The 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 involving three decisions, as opposed to two decisions, but otherwise follows the same spirit.

Our main contribution (Theorem 2.4) is to show that, w.h.p. as the size of the instance diverges to infinity, every "balanced" sequential local algorithm fails to produce an ℓ-satisfying assignment when the ratio d of the number of clauses to the number of variables exceeds $(1 + o_K(1))\frac{2^{K-1}}{K} \ln^2 K$, clause size K is sufficiently large (but independent from the number of variables), and $r = O((\ln \ln n)^{O(1)})$. Specifically, we will show this when the ratio of ℓ to the total number of clauses dn is below a certain constant less than unity (see the aforementioned theorem for details). "Balance" is a technical condition explained in Definition 2.3, which says that the local algorithm respects the inherent symmetry between 0 and 1. It is a condition satisfied by all known algorithms including BP- and SP-guided decimation, as we establish.

Our bound on the ratio d is reasonably close to bounds at which simple algorithms actually work. In particular, it is well known that a very simple Unit Clause algorithm is capable of finding satisfying assignments for this problem when d is below $\rho\frac{2^{K-1}}{K}$ for some universal constant ρ [AKKT02] for K sufficiently large. The Unit Clause algorithm is the best known algorithm for this problem. (A better algorithm is known for the random K-SAT problem that works up to clause-to-variables ratio $(1 - o_K(1))\frac{2^K}{K} \ln K$ [CO10]. It is likely that a similar idea can be applied to the NAE-K-SAT setting, but such a result is not available, to the best of our knowledge.) One of the hopes was that BP- and SP-guided decimation algorithms might be able to bridge this factor of K between the Unit Clause algorithm and the satisfiability threshold d_s above. Our result, however, implies that, short of possibly a $\ln^2 K$ multiplicative factor, the "infamous" factor-$O(K)$ gap between the satisfiability threshold and the region achievable by known algorithms cannot be broken by means of sequential local algorithms, in particular by BP- and SP-guided decimation algorithms with $O(r)$ number of rounds of message-passing iterations.

Previously, Coja-Oghlan [CO11] showed that the BP-guided decimation algorithm fails to find satisfying assignments for random K-SAT problems when $d \geq \rho\frac{2^K}{K}$ for some universal constant ρ, for an arbitrary number of iterations r, which in particular might depend on the number of variables. (Here 2^K factor is an "appropriate" substitution for 2^{K-1} when switching from NAE-K-SAT to the K-SAT problem. We maintain this distinction, even though technically it is eliminated by constant ρ.) It is reasonable to expect that that result holds also for the NAE-K-SAT problem using the same analysis. Thus our result partially reproduces the main result of [CO11] in the special case when the number of iterations is bounded by $O((\ln \ln n)^{O(1)})$ (short of an additional $\ln^2 K$ factor). At the same time, however, our result is applied in a "blanket way" to a broad class of algorithms, including most notably an SP-guided decimation algorithm with the number of iterations bounded by the same value, and our analysis is insensitive to the details of the algorithm.

Since the first version of our paper was posted, we have become aware of the result by Hetterich [Het16], which shows that the SP-guided decimation algorithm w.h.p. fails to find a satisfying assignment of a random K-SAT formula above density $(d_s/K) \ln K$ for $d_s = 2^K \ln 2$ for all sufficiently large K. That result, unlike ours, does not assume any bound on the number of iterations of the SP-guided decimation algorithm and applies to a slightly smaller formula density $(d_s/K) \ln K$, as opposed to density $(d_s/K) \ln^2 K$ appearing in our main result.

1.2. Techniques. Our main proof technique relies on the intricate geometry of the solution space of the random NAE-K-SAT problem. Specifically it relies on the so-called *m-overlap* structure of nearly satisfying assignments of random NAE-K-SAT, which relates to the space of possible pairwise Hamming distances between m

such satisfying assignments. Previously this overlap structure was studied for the case $m = 2$ for the random K-SAT problem and several other related problems, including the problem of proper coloring of sparse random graphs [ACORT11], [ACO08]. A certain *shattering* property was established which, roughly speaking, says that above a certain density, the Hamming distance between every pair of satisfying assignments (overlap), normalized by the number of variables, is either smaller than a certain constant δ_1 or larger than some constant $1 \geq \delta_2 > \delta_1$. As as result, the solution space can be partitioned into different subsets (clusters) based on their proximity to each other. For the case of the NAE-K-SAT problem this 2-overlap property can be established for densities d approximately $d > d_s/2$. (A weaker version of this result corresponding to "almost" all pairs does hold at densities above $O\left(\frac{d_s}{K} \ln K\right)$ [MRT11].) Unfortunately, this is not strong enough to cover the regime of $d > (d_s/K) \ln^2 K$ claimed in our main theorem, so instead we have to establish a certain property regarding m-overlaps of satisfying assignments for appropriately chosen $m > 2$. This is the essence of Theorem 4.1, which we prove in this paper. Roughly speaking, this theorem says that, with probability at least $1-\exp(-\Omega(n))$, when $d \geq (1+\epsilon)\frac{d_s}{K} \ln^2 K$, and K is sufficiently large, one cannot find $m \approx \epsilon K / \ln K$ satisfying assignments such that the Hamming distance (overlap) between every pair of the assignments normalized by the number of variables is $\approx \ln K/K$. The result applies to ℓ-satisfying assignments as well for sufficiently small $\ell < dn$. Then for every $\beta \in (0, 1)$ we establish the following result. If a sequential local algorithm is capable of finding an ℓ-satisfying assignment, with probability at least $n^{-(\ln \ln n)^{O(1)}}$, then by running the algorithm m times and constructing a certain interpolation scheme, one can construct m ℓ-satisfying assignments such that the pairwise normalized distance between any pair of these assignments is $\approx \beta$. This contradicts Theorem 4.1. Our superpolynomial upper bound $n^{-(\ln \ln n)^{O(1)}}$ on the likelihood of success also rules out the possibility of running the algorithm for polynomially many independent trials in the hope of finding at least one ℓ-satisfying assignment.

The link between the overlap property and the ensuing demise of local algorithms was recently established by authors [GS14] in a different context of finding a largest independent set in a random regular graph. There the argument was used to show that so-called i.i.d. factor-based local algorithms are incapable of finding nearly largest independent sets in random regular graphs, refuting an earlier conjecture by Hatami, Lovász, and Szegedy [HLS]. The result was further strengthened by Rahman and Virag [RV14], who obtained essentially the tightest possible result, using m-overlap structures of "large" independent sets. Our use of m-overlaps is inspired by this work, though the set of restrictions on the m-overlaps implied by Theorem 4.1 is much simpler than that appearing in [RV14].

An important technical and conceptual difference between the present work and that of [GS14] and [RV14] is that algorithms considered in the aforementioned papers are not sequential. Instead the decision taken by each variable in those models is taken simultaneously for all variables. In the case of sequential local algorithms, since the variables are set sequentially, the decision for one variable can be nonlocalized for the remaining variables, thus creating potential long-range dependencies. We deal with this potential long-range impact of decisions as follows. We associate variables with random i.i.d. weights chosen from an arbitrary continuous distribution, for example a uniform distribution. The weights are used solely to determine the order of fixing the values of the variables during the progression of the sequential local algorithm. Specifically the largest weight first rule is used. The decision to fix the value of a

particular variable then can only impact variables with lower weights. Specifically if the value of variable x is fixed now, the value of variable y can be impacted *only* if there exists a sequence $x_0 = x, x_1, \ldots, x_\ell = y$ such that the distance between x_i and x_{i+1} is at most r (the radius of the decision making rule) and the weight of x_i is larger than that of x_{i+1} for all i. For a given set of variables $x_0, \ldots, x_\ell$ the likelihood of this total order of variables is $1/\ell!$, which decays faster than any exponential function in ℓ. This, coupled with the fact that the growth rate of nodes at distance at most $r\ell$ from x is at most exponential in $r\ell$, will allow us to control the range of influence of the variable x when its value is set. A similar idea of controlling the range of influence is used in the analysis of local algorithms in several places, including [NO08]. Bounding the ranges of influence is a crucial idea in implementing the interpolation scheme and constructing m assignments with "nonexistent" normalized overlaps β.

1.3. Contrast with empirical studies of SP-guided decimation. Our study of local sequential algorithms is motivated in part by an attempt to understand the analytic behavior of some "natural," statistical-physics-motivated, algorithms for constraint satisfaction problems on random instances. These algorithms, specifically BP-guided and SP-guided decimation algorithms, exhibit a spectacular performance empirically, capable of finding solutions very rapidly and very close to the thresholds, beyond which the satisfying assignments do not exist or are conjectured not to exist. A partial list of references documenting the performance of these algorithms includes the papers [MPZ02], [BMZ05], [KMRT+07], [RTS09], [DRZ08], [KSS12] as well as the book by Mezard and Montanari [MM09]. At the same time, mathematically rigorous analysis of these algorithms is mostly lacking. Notable exceptions are the works of Coja-Oghlan [CO11] and Hetterich [Het16], who analyzed the performance of the BP- and SP-guided decimation algorithm for the random K-SAT problem, and which we have discussed above, and of Maneva, Mossel, and Wainwright [MMW07], who reformulate the SP algorithm as the BP algorithm on a "lifted" Markov random field.

The literature on BP- and especially SP-guided decimation (for instance, [BMZ05], [KMRT+07], [RTS09], [MM09]) has shown that these algorithms perform well empirically on random instances of K-SAT for small values of K ($K \leq 10$). There are several ways in which these implementations differ (or may differ) from the setting we study: (1) They analyze K-SAT, as opposed to NAE-K-SAT, and the asymmetry in K-SAT may already make a difference for the algorithm. (2) They study 3-SAT, so very local constraints, while we study K-clauses where K is constant but large, and this increase in the locality of the constraints may make it harder for local algorithms to function effectively (even though the locality of the algorithm can be chosen to be arbitrarily after K is fixed). (3) In the empirically analyzed algorithms, the order in which variables are set is not fixed a priori, but may depend on the probability estimates returned by the message-passing iterations. While this could possibly also affect the ability of the algorithms to find satisfying assignments, there appears to be no reason based on the statistical-physics theory which implies that such a presorting of variables is a crucial for SP-guided decimation algorithm to succeed. Size biasing rather appears to be a sensible implementation detail of the algorithm. (Some discussion of the accuracy of the size-biased version versus random order can be found in [KSS12].) (4) Finally, and probably most significantly, we analyze algorithms that work with only a moderately growing number of rounds of message-passing iterations, and this allows us to fit our approach within the framework of sequential local algorithms. In contrast the empirical studies suggest using message passing till the iterations converge, and this may take more than linear or even exponential number

of rounds. This gap, however, was recently closed in [Het16], as previously mentioned.

Thus our work and setting make a collection of choices that are different from some of the earlier works in the hope of getting some formal analysis. Unfortunately our results show that, when the four choices are combined, it definitely produces a provable difference, and the algorithms fail to find satisfying assignments at densities that are qualitatively below the satisfiability threshold. Of course, it would be important to reduce the number of parameters in which the choices for the negative results differ from those used in the empirical setting (which yielded positive results), and we hope this will be a subject of future work.

1.4. Future work. Going beyond specific classes of algorithms, a major challenge is to understand the intrinsic complexity of finding satisfying assignments in random instances of K-SAT and NAE-K-SAT problems. Given the repeated failure to produce polynomial time algorithms for, say NAE-K-SAT, above the density threshold of $(1+o_K(1))\frac{2^{K-1}}{K}\ln^2 K$, it is plausible that the problem is actually average-case hard in this regime. The formalism of problems which are NP-hard on average is available [Lev86]; however, the problems which are known to be hard on average are not particularly natural and are quite distant from the types of problems considered here. Another problem that has defied designing a fast algorithm, and which is closer in spirit to the problems considered in this and related papers, is the problem of finding a largest independent set in a dense random graph. Specifically, consider the graph $\mathbb{G}(n, 1/2)$, where every one of the $n(n-1)/2$ undirected edges is present with probability $1/2$ independently for all edges. It is known that the largest independent set has size $2(1+o(1))\ln_2 n$ w.h.p. At the same time the best known algorithm (greedy) finds only an independent set of size $(1+o(1))\ln_2 n$, and bridging this gap has been a major open problem in the field of combinatorics and random graphs since Karp posed it as an open problem back in 1976 [Kar76]. It is entirely plausible that this problem is NP-hard in the constant average-degree case (i.e, on the random graph $\mathbb{G}(n, d/n)$ for constant d), and resolving this question one way or the other is a major open problem in theoretical computer science. By drawing an analogy with this, and in light of 40 years of repeated failure to produce an algorithm for this problem, it is plausible to conjecture that NAE-K-SAT and related problems are NP-hard on average above thresholds corresponding to the emergence of nontrivial restricted overlap properties, similar to the ones established in this paper. Shedding some light on this question is perhaps one of the most interesting problems in the area of random constraint satisfaction problems.

Organization and notational conventions. Our 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 property regarding m-overlaps of satisfying assignments is established in section 4. The proof of the main result is in section 5.

Throughout the paper we use standard order-of magnitude-notations $O(\cdot), o(\cdot)$ for sequences defined in terms of the number of Boolean variables n. The constants hidden by this notation may depend on any other parameters of the model, such as K and d. Similarly we use notations $O_K(\cdot)$ and $o_K(\cdot)$ to denote sequences indexed by K as $K \to \infty$. The constants hidden in these notations are universal.

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.

2.1. The 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 the NAE-K-SAT problem is described as a collection of n binary variables $x_1, \dots, x_n$ taking values 0 and 1 and a collection of N clauses $C_1, \dots, C_N$, where each clause is given by a subset of K literals. Each literal is a variable x in $x_1, \dots, x_n$ or negation $\bar{x}$ of a variable. An assignment is a function $\sigma : \{x_1, \dots, x_n\} \to \{0, 1\}$. σ satisfies a clause C_j if in this clause there are at least one literal valued 1 and at least one literal valued 0. σ satisfies the formula Φ if every clause C_j, $1 \le j \le m$, is satisfied. For every assignment $\sigma = (\sigma(x_i), 1 \le i \le n)$, let $\bar{\sigma} = 1 - \sigma$ be the assignment given by $\bar{\sigma}(x_i) = 1 - \sigma(x_i)$, $1 \le i \le n$. Given a formula Φ, denote by $\mathbb{SAT}(\Phi) \subset \{0, 1\}^n$ the (possibly empty) set of satisfying assignments σ. For every $0 \le \ell \le m$, denote by $\mathbb{SAT}(\Phi, \ell) \supset \mathbb{SAT}(\Phi)$ the set of assignments violating at most ℓ clauses, so that $\mathbb{SAT}(\Phi, 0) = \mathbb{SAT}(\Phi)$. Every $\sigma \in \mathbb{SAT}(\Phi, \ell)$ will be called an ℓ-satisfying assignment (or simply satisfying assignment when $\ell = 0$). 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, every ℓ, and assignment σ, we have that σ ℓ-satisfies Φ if and only if $\bar{\sigma}$ ℓ-satisfies Φ. Consequently, suppose $\mathbb{SAT}(\Phi, \ell) \neq \emptyset$. Then if σ is drawn uniformly at random from $\mathbb{SAT}(\Phi, \ell)$, for every $1 \le i \le n$ we have*

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

Reduced instances. We now introduce some notation for *reduced* instances of NAE-K-SAT. Informally, "reduced" instances are obtained from normal instances of NAE-K-SAT by giving a partial assignment to some of the variables. Formally, a clause of a reduced instance C is given by a set of at most K literals, along with a sign denoted $\mathrm{sign}(C) \in \{+, -, 0\}$. Furthermore, C has exactly K literals if and only if $\mathrm{sign}(C) = 0$. (Sometimes we refer to these signs as decorations.) An assignment σ satisfies a reduced clause C if one of the following takes place: $\mathrm{sign}(C) = +$ and some literal in C is assigned 0 by σ, OR $\mathrm{sign}(C) = -$ and some literal in C is assigned 1 by σ, OR $\mathrm{sign}(C) = 0$ 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 ℓ-satisfies Φ if it violates at most ℓ clauses in Φ. A partial assignment $\sigma : \{x_{n+1}, \dots, x_{n+t}\} \to \{0, 1\}$ reduces a (reduced) NAE-K-SAT instance Φ on variables $x_1, \dots, x_{n+t}$ to a reduced instance Ψ on variables $x_1, \dots, x_n$ in a natural way, so that an assignment $\tau : \{x_1, \dots, x_n\} \to \{0, 1\}$ ℓ-satisfies Ψ if and only if the combined assignment $\tau \circ \sigma$ ℓ-satisfies Φ.

Note that Observation 2.1 does not necessarily hold for the reduced instances of the NAE-K-SAT problem. Instances in which every clause has sign 0 will be called nonreduced 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 $\mathrm{sign}(C) = +$, then $\mathrm{sign}(\bar{C}) = -$; if $\mathrm{sign}(C) = -$, then $\mathrm{sign}(\bar{C}) = +$; and if $\mathrm{sign}(C) = 0$, then $\mathrm{sign}(\bar{C}) = 0$. 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 a reduced instance Φ on variables $x_1, \dots, x_n$ and a reduced instance Ψ on variables $x_1, \dots, x_{n+t}$, suppose that Φ is the instance derived by reducing Ψ with the assignment $\sigma : \{x_{n+1}, \dots, x_{n+t}\} \to \{0, 1\}$. Then $\bar{\Phi}$ is the reduced*

instance obtained by reducing $\bar{\Psi}$ *with the assignment* $\bar{\sigma}$*, where* $\bar{\sigma}(x_i) = 1 - \sigma(x_i)$.

In particular, whenever a reduced formula Φ is obtained from a nonreduced formula Ψ by setting some variables of Ψ, setting the same variables to opposite values generates the complement $\bar{\Phi}$ of Φ.

Random NAE-K-SAT problem. We denote by $\mathbf{\Phi}(n, dn)$ a random (nonreduced) instance of the NAE-K-SAT problem on variables $x_1, \ldots, x_n$ and $\lfloor dn \rfloor$ clauses $C_1, \ldots, C_N$ 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 $j = 1, 2, \ldots, N$. 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. d is called the clauses-to-variables ratio or the density of the formula.

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 undirected graph with left nodes corresponding to the variables and right nodes corresponding to the clauses. A clause node is connected to a variable node if and only if this variable appears in this clause. The edges are labeled positive or negative to indicate the polarity of the literal in the clause. In the case when Φ is a reduced NAE-K-SAT instance, clause vertices are also labelled with the sign of the clause. Thus the factor graph of an NAE-K-SAT instance uniquely defines this instance.

The second graph that 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 and only if they appear in the same clause. 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 an 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 $B(x, r)$ alone. Since r is even, we have that the factor graph of this induced instance is $B(x, r)$.

2.2. Sequential local algorithms for the NAE-*K*-SAT problem and the main result. We now define the notion of sequential local algorithms formally and state our main result.

Fix a positive even integer $r \geq 0$. In our setting r will depend on model parameters such as the number of variables n in a random formula $\mathbf{\Phi}(n, dn)$. Denote by $\mathcal{SAT}_r$ the set of all NAE-K-SAT reduced and nonreduced instances Ψ with a designated (root) variable x such that the distance from x to any other variable in Ψ is at most r in $\mathbb{F}(\Psi)$. We note that $\mathcal{SAT}_r$ is an infinite set. $\mathcal{SAT}_r$ is the set of all instances Ψ which can be observed as depth r neighborhood $B_\Phi(x, r)$ of an arbitrary variable x in an arbitrary reduced and nonreduced NAE-K-SAT instance Φ.

Consider any function $\tau : \mathcal{SAT}_r \to [0, 1]$ which takes as an argument an arbitrary member $\Psi \in \mathcal{SAT}_r$ and outputs a value (probability) in $[0, 1]$. We now describe a sequential local algorithm, which we refer to as the τ*-decimation algorithm*, for solving the NAE-K-SAT problem. Given a positive even integer r, the depth-r neighborhood $B(x_i, r) = B_{\mathbf{\Phi}(n,dn)}(x_i, r) \in \mathcal{SAT}_r$ of any fixed variable $x_i \in [n]$ in the

formula $\mathbf{\Phi}(n, dn)$, rooted at x_i, 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 $x_1, \ldots, x_n$ 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 nonreduced instance $B(x_i, r)$, according to the underlying local algorithm. Specifically, we now describe how the function τ is used as a basis of a local algorithm to generate an assignment $\sigma : \{x_1, \ldots, x_n\} \to \{0, 1\}$.

τ-decimation algorithm.

INPUT:
an instance Φ of an NAE-K-SAT formula on binary variables $x_1, \ldots, x_n$,
a positive even integer r and function τ.

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 Φ_i to be the reduced instance obtained from Φ_{i-1} by fixing the value of x_i as above, removing satisfied and violated clauses, and decorating newly generated partially satisfied clauses with $+$ and $-$ appropriately.

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

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 removing violated clauses from the formula. We denote by $\sigma_{\Phi,\tau}$ the (random) output $\sigma(x_1), \ldots, \sigma(x_n)$ produced by the τ-decimation algorithm above. We say that the τ-decimation algorithm solves instance Φ if the output $\sigma_{\Phi,\tau}$ is a satisfying assignment, namely $\sigma_{\Phi,\tau} \in \mathbb{SAT}(\Phi)$. Similarly, we say that the τ-decimation algorithm ℓ-solves instance Φ if $\sigma_{\Phi,\tau}$ violates at most ℓ clauses. We now 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 $\tau(\bar{\Phi}) = 1 - \tau(\Phi)$.*

The balance condition above basically says that the τ-decimation algorithm does not have a prior bias in setting variables to 1 versus 0. In particular, when the instance is nonreduced, the τ-decimation algorithm sets variable values equi-probably, consistent 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. For every $\epsilon > 0$, let

$$\kappa = \kappa(\epsilon, K) = (2\ln(2))^{-1}\epsilon^3 \ln^2 K/K^2. \tag{1}$$

THEOREM 2.4. *For every $\epsilon > 0, \xi \in (0, 1)$ there exists K_0 such that for every $K \geq K_0$, $d > (1 + \epsilon)2^{K-1} \ln^2 K/K$, every even $r \leq (\ln \ln n)^{1-\xi}$, and every balanced local rule $\tau : \mathcal{SAT}_r \to [0, 1]$ the following holds:*

$$\mathbb{P}\left(\sigma_{\mathbf{\Phi}(n,dn),\tau} \in \mathbb{SAT}(\mathbf{\Phi}(n, dn), \kappa n)\right) \leq \exp\left(-\ln n(\ln \ln n)^{\xi/7}\right)$$

for all large enough n.

Namely, with high probability (asymptotically at least $1-\exp(-O(\ln(\ln\ln n))^{O(1)})$, every τ-decimation algorithm will violate at least $O\left(\frac{\ln^2 K}{K^2}n\right)$ clauses. As we have mentioned above, the threshold for satisfiability is $d_s = 2^{K-1}\ln 2 - \ln 2/2 - 1/4 - o_K(1)$. Thus our theorem implies that sequential local algorithms fail to find even a near satisfying assignment at densities approximately $(d_s/K)\ln^2 K$. We note that a superpolynomial decay rate $\exp(-O(\ln(\ln\ln n))^{O(1)})$ prevents the possibility of finding a good solution by rerunning a τ-decimation algorithm polynomially many times.

2.3. BP-guided and SP-guided decimation algorithms as local sequential algorithms. We now show that BP-guided decimation and SP-guided decimation algorithms are in fact special cases of τ-decimation algorithms, as described in the previous section, when the number of message-passing iterations is at most $(\ln\ln n)^{1-\xi}$. As a consequence we have that the negative result given by 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 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 bipartite factor graph $\mathbb{F}(\Phi(n, dn))$. After several rounds of such exchanges 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 for only r rounds, then for every variable x_i such that the neighborhood $B(x_i, r)$ is in fact a tree, 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. A standard fact is that, for the majority of variables, $B(x_i, r)$ is indeed a tree even up to $r \le O(\ln n)$ for an appropriate constant hidden in $O(\cdot)$. Thus most of the time 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 the 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 to $\sigma(x_1) = 0$ with probability $1/(\mu(x_1)+1)$. Namely, the variable is set probabilistically proportionally to the ratio of the number of solutions setting it to one versus the number of solutions setting it to zero. After the decision for variable x_1 is set in the way described above, the variable x_2 is selected from the reduced formula on variables $x_2, \ldots, x_n$. 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 a BP-guided decimation algorithm. It is thus parametrized by the computation depth r.

It is clear that such a BP-guided decimation algorithm is precisely a τ-decimation algorithm, where $\tau(B(x_i, r)) = \mu(x_i)/(\mu(x_i)+1)$ is the marginal probability of the variable x_i corresponding to the reduced formula $B(x_i, r)$. Furthermore, the τ-local

rule so obtained satisfies the balance condition described in Definition 2.3. Thus, as an implication of our main result, Theorem 2.4, we conclude that the BP-guided decimation algorithm fails to find a satisfying assignment for $\Phi(n, dn)$ in the regime where our result on the τ-decimation algorithms applies. Let κ be defined by (1).

COROLLARY 2.5. *For every $\epsilon > 0$, $\xi \in (0,1)$ there exist K_0 and $\delta > 0$ such that for every $K \geq K_0$, $d > (1+\epsilon)2^{K-1}\ln^2 K/K$, and every even $r \leq (\ln\ln n)^{1-\xi}$ the following holds:*

$$\mathbb{P}\,(\textit{BP-guided decimation algorithm with } r \textit{ iterations violates at most } \kappa n \textit{ clauses}) \leq \exp\left(-\ln n(\ln\ln n)^{\xi/7}\right)$$

for all large enough n.

Survey propagation. We now describe the SP-guided decimation algorithm. The algorithm is significantly more complex to describe, but we will show again that it is a τ-decimation algorithm when the number of message-passing rounds is bounded by $r \leq (\ln\ln n)^{1-\xi}$, and that τ is a balanced rule. As a consequence we will conclude that the SP-guided decimation algorithm also fails to find satisfying assignments for instances with density larger than $(d_s/K)\ln^2 K$ when the number of rounds is bounded by a constant. This is summarized in Corollary 2.7 below. The details of the algorithm are delayed till the appendix. The main implication of this discussion is the following fact, the proof of which is also found in the appendix.

OBSERVATION 2.6. *The local rule τ corresponding to the SP iterations is balanced.*

Theorem 2.4 then becomes applicable, and we conclude that, letting κ be defined by (1), the following holds.

COROLLARY 2.7. *For every $\epsilon > 0$, $\xi \in (0,1)$ there exist K_0 and $\delta > 0$ such that for every $K \geq K_0$, $d > (1+\epsilon)2^{K-1}\ln^2 K/K$, and every even $r \leq (\ln\ln n)^{1-\xi}$ the following holds:*

$$\mathbb{P}\,(\textit{SP-guided decimation algorithm with } r \textit{ iterations violates at most } \kappa n \textit{ clauses}) \leq \exp\left(-\ln n(\ln\ln n)^{\xi/7}\right)$$

for all large enough n.

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 result is simple to state—we show that balanced local rules lead to unbiased decisions for *every* nonreduced 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 reduced or nonreduced 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 an appropriately 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 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 3.5 formalize these claims.

In what follows, we first introduce some notation that makes the decisions of our 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 distribution uniform in $[0, 1]$, independent from the random formula $\mathbf{\Phi}(n, dn)$. Let $\pi : [n] \to [n]$ be the permutation induced by the order statistics of $\mathbf{Z}$. Namely, $Z_{\pi(1)} > Z_{\pi(2)} > \cdots > Z_{\pi(n)}$. We now 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. Namely, we assume that the τ-decimation algorithm performed on a random instance of the NAE-K-SAT problem $\mathbf{\Phi}(n, dn)$ is conducted 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 $\mathbf{\Phi}$ 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-decimation 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 $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_i)$, $1 \leq i \leq n$, the output of the τ-decimation 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 $B_{\Phi,\mathbf{z},\mathbf{u}}(x_i, r)$, $1 \leq i \leq n$, 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 Definition 2.3 of balanced local rules.

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

$$B_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_i, r) = \bar{B}_{\Phi,\mathbf{z},\mathbf{u}}(x_i, r), \tag{2}$$

$$\sigma_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_i) = 1 - \sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_i), \tag{3}$$

where $\bar{\mathbf{u}}$ is defined by $\bar{u}_i = 1 - u_i$, $1 \leq i \leq n$. 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

$i = 1, 2, \ldots, n$:

$$\mathbb{P}(\sigma_{\Phi,\mathbf{z},\mathbf{U}}(x_i) = 0) = 1/2. \tag{4}$$

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 (3) holds for all variables $x_{\pi(j)}$, $j \le i-1$ (namely, for variables whose values are already determined at time i), and the identity (2) 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 nonreduced 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 (2) is verified.

Fix $i \ge 1$, and assume now that the inductive hypothesis holds for $j \le i$. In particular, the values $\sigma(x_{\pi(j)})$ determined for $j = 1, \ldots, i-1$ under $\mathbf{u}$ and $\bar{\mathbf{u}}$ satisfy (3). Now consider the step of assigning the value of $x_{\pi(i)}$. We have $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)}) = 1$ if and only if $u_{\pi(i)} \le \tau(B_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)}, r))$, and $\sigma_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_{\pi(i)}) = 1$ if and only if $\bar{u}_{\pi(i)} \le \tau(B_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_{\pi(i)}, r))$. By the inductive assumption we have that $B_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_{\pi(i)}, r) = \bar{B}_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)}, r)$. Since τ is balanced, 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 $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)}) = 1$ if and only if $\sigma_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_{\pi(i)}) = 0$. Namely, $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)}) = 1 - \sigma_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_{\pi(i)})$, and identity (3) is verified.

It remains to show that identity (2) 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 that this neighborhood contains one or several clauses which contain $x_{\pi(i)}$. 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 (3) 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 (2) 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 that $\sigma_{\Phi,\mathbf{z},\bar{\mathbf{u}}}(x_{\pi(i)})$ 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 (2) is verified.

The case when clause C is signed − under $\mathbf{u}$ and signed + under $\bar{\mathbf{u}}$ is considered similarly. Finally, suppose that $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_{\pi(i)})$ 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 the 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 (4). ∎

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 random choices of the τ-decimation algorithm). With some abuse of notation, we write Z_x for the random label associated with variable x. In particular, $Z_{x_{\pi(1)}} = Z_{\pi(1)}$, $Z_{x_{\pi(2)}} = Z_{\pi(2)}$, etc. We introduce the following relationship between the variables $x_1, \dots, x_n$ of our formula.

DEFINITION 3.2. *Given a formula Φ and random sequence $\mathbf{z} = (z_i, 1 \le i \le n)$, we say that x_i* influences *x_j if either $x_j = x_i$ or in the underlying node-to-node graph $\mathbb{G} = \mathbb{G}(\Phi)$ there exists a sequence of nodes $y_0, y_1, \dots, y_t \in \{x_1, \dots, x_n\}$ 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 $l = 0, 1, \dots, t-1$.*
(iii) *$Z_{y_{l-1}} > Z_{y_l}$ for $l = 1, 2, \dots, t$. 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 the 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 $\rightsquigarrow$ are functions of the randomness of the formula $\mathbf{\Phi}(n, dn)$ and vector $\mathbf{Z}$, but independent from the random vector $\mathbf{U}$.

While the definition above is sound for every $r > 0$, we will apply it in the case where r is the parameter appearing in the context of the τ-decimation algorithm, namely, in the context of the function τ defined on the set of rooted instances $\mathcal{SAT}_r$ introduced 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 $\mathbf{Z}$, respectively, suppose $\mathbf{u} = (u_i, 1 \le i \le n)$ and $\mathbf{u}' = (u'_i, 1 \le i \le n)$ are such that $u_{i_0} \ne u'_{i_0}$ and $u_i = u'_i$ for all $i \ne i_0$ for some fixed index 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 value of $\mathbf{u}$ at i_0 may impact the decisions associated only with variables x influenced by x_{i_0}.*

Proof. Fix any variable x_i such that $\sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_i) \ne \sigma_{\Phi,\mathbf{z},\mathbf{u}'}(x_i)$. If $i = i_0$, then trivially $x_{i_0} \rightsquigarrow x_{i_0}$ and thus $x_{i_0} \in \mathcal{IR}_{x_{i_0}}$. Otherwise assume $i \ne i_0$ and thus $u'_i = u_i$. Then it must be the case that $\tau(B_{\Phi,\mathbf{z},\mathbf{u}'}(x_i, r)) \ne \tau(B_{\Phi,\mathbf{z},\mathbf{u}}(x_i, r))$, since otherwise with the same value of $u_i = u_{i'}$ we would have the same assignment: $\sigma_{\Phi,\mathbf{z},\mathbf{u}'}(x_i) = \sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_i)$. This implies that there exists 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, namely, $\sigma_{\Phi,\mathbf{z},\mathbf{u}'}(x_{i_1}) \ne \sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_{i_1})$. Then either $i_1 = i_0$ and in particular $x_{i_0} \rightsquigarrow x_{i_1}$, or again $\tau(B_{\Phi,\mathbf{z},\mathbf{u}'}(x_{i_1}, r)) \ne \tau(B_{\Phi,\mathbf{z},\mathbf{u}}(x_{i_1}, r))$, further implying the existence of a variable x_{i_2} with distance at most r from x_{i_1} such that $z_{x_{i_2}} > z_{x_{i_1}}$ and $\sigma_{\Phi,\mathbf{z},\mathbf{u}'}(x_{i_2}) \ne \sigma_{\Phi,\mathbf{z},\mathbf{u}}(x_{i_2})$. Continuing this reasoning, we will eventually arrive at node i_0, implying that $x_{i_0} \rightsquigarrow x_i$ and completing the proof. ∎

We now obtain a probabilistic bound on the size of a largest in cardinality influence range class $\mathcal{IR}_{x_i}$, $1 \le i \le n$. Fix a variable x in $\mathbf{\Phi}(n, dn)$. First we obtain a probabilistic bound on the size of a neighborhood $B(x, t)$ around x for appropriately small values of t.

LEMMA 3.4. *For every* $0 < \epsilon < 1/5$, $\delta = \epsilon/(4\ln(edK))$, *and* $t = \lfloor \delta \ln n \rfloor$,

$$\mathbb{P}(|B(x,t)| \geq n^{\epsilon}) \leq \exp\big(-n^{\epsilon/5}\big) \tag{5}$$

for all sufficiently large n.

Proof. For the proof it will be convenient to switch to a model of $\mathbf{\Phi}(n, dn)$ in which the number of clauses is distributed as a Poisson random variable with mean $(1+\alpha)dn$ for a fixed $\alpha > 0$, rather than taking the precise value $\lfloor dn \rfloor$. We denote such a random formula by $\hat{\mathbf{\Phi}}(n, dn, \alpha)$. By a straightforward concentration result, we have that the number of clauses in $\hat{\mathbf{\Phi}}(n, dn, \alpha)$ exceeds dn with probability at least $1 - \exp(-\gamma n)$ for some γ which depends on α, K, and d. Since the bound in (5) is of the form $\exp(-n^{\epsilon/5})$, and $\epsilon/5 < 1$, it suffices to establish a bound on $B(x,t)$ for $\hat{\mathbf{\Phi}}(n, dn, \alpha)$ instead. For notational convenience we assume that the mean number of clauses is dn as opposed to $(1+\alpha)dn$. It will be easy to see that the argument does not really depend on the actual value of d as long as it is constant.

Next we use a standard approximation of $B(x,t)$, $t = 1, 2, \ldots, t$, by a branching process with outdegree distribution given by a Poisson random variable with mean dK. In fact we will use the property that such a branching process stochastically dominates $B(x,t)$, for *every* t, which we now establish. Towards this goal, we introduce the following revelation process, which is again a standard method of analysis of neighborhoods of a node in a random graph. In each step of the revelation process, the nodes $[n]$ are partitioned into three groups: "dead," "alive," and "unexplored," denoted respectively by $D_k, A_k, I_k, k \geq 0$. For $k = 0$ we set $D_0 = \emptyset$, $A_0 = \{x\}$, $I_0 = [n] \setminus \{x\}$. Assuming the sets $D_{k-1}, A_{k-1}, I_{k-1}$ are defined, we define the sets with index k as follows. If $A_{k-1} = \emptyset$, we set $D_k = D_{k-1}$, $A_k = A_{k-1}$, $I_k = I_{k-1}$, and the revelation process stops. Otherwise, we pick an arbitrary node $y \in A_{k-1}$ and let $y_1, \ldots, y_\Delta \in I_{k-1}$ be the neighbors of y in I_{k-1} with respect to graph $\mathbb{G}(\mathbf{\Phi}(n, dn))$. Then we set $D_k = D_{k-1} \cup \{y\}$, $A_k = A_{k-1} \cup \{y_1, \ldots, y_\Delta\} \setminus \{y\}$, and $I_k = I_{k-1} \setminus \{y_1, \ldots, y_\Delta\}$. The process $D_k, A_k, I_k, k \geq 0$, can be viewed as a branching process with root x, where in every step one of nodes y of the tree is chosen and its children $y_1, \ldots, y_\Delta$ are revealed. Conditional on sets $D_{k-1}, A_{k-1}, I_{k-1}$, Δ is distributed as a Poisson random variable with mean at most dK. Thus the Poisson branching process with mean $\beta \triangleq dK \geq 1$ stochastically dominates $B(x,t)$ for all t, and it suffices to obtain a bound on the number of nodes in the first t generations of the Poisson process with mean dK. For notational convenience, the Poisson branching process is also denoted by $B(x,t)$. More specifically, letting W_ℓ, $\ell \geq 0$, denote the number of nodes in the ℓth generation of this process, with $W_0 = 1$ corresponding to the root x, we have $|B(x,t)| = \sum_{0 \leq \ell \leq t} W_\ell$. In particular W_1 has a Poisson distribution with mean β.

We claim that the following upper bound holds for each $l \leq t = \lfloor \delta \ln n \rfloor$,

$$\mathbb{P}(W_l > n^{\epsilon/2}) \leq \exp(-n^{\epsilon/4} + o(n^{\epsilon/4})), \tag{6}$$

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 $G(\theta) = \mathbb{E}[\theta^{W_1}]$ for $\theta > 0$, where we recall that W_1 has a Poisson mean β distribution. Then $G(\theta) = \exp(\beta\theta - \beta)$ and $\mathbb{E}[\theta^{W_l}] = G^{(l)}(\theta)$ is the lth iterate of function $G(\theta)$. Let $\theta = 1 + \frac{1}{(e\beta)^t}$. Define $\gamma_l = 1/(e\beta)^l$, $0 \leq \ell \leq t$. 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 that $\beta > 1$ implies $\beta\gamma_t < 1$, and inequality $e^z \le 1 + ez$ for $z \le 1$. Then

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

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

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

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 $t \le \delta \ln n$, then $\gamma_t \ge (e\beta)^{-\delta \ln n} = n^{-\ln(e\beta)\delta}$, implying the upper bound $\exp(-n^{\epsilon/4} + o(n^{\epsilon/4}))$, by the choice of δ. This completes the proof of the bound (6) and of the lemma. ∎

We now return to obtaining bounds on the sizes of the influence ranges $\mathcal{IR}_{x_i}$.

PROPOSITION 3.5. *For every* $0 < \epsilon < 1/5$,

$$\mathbb{P}\left(\max_{1\le i\le n} |\mathcal{IR}_{x_i}| \ge n^{\epsilon}\right) \le \exp\left(-\ln n(\ln\ln n)^{\xi/4}\right) \tag{7}$$

for all large enough n.

Proof. Similarly to the proof of Lemma 3.4, it will be convenient to switch to an equivalent model where instead of generating $\lfloor dn \rfloor$ clauses uniformly at random for the formula $\mathbf{\Phi}(n, dn)$, each of the total universe of $2^K\binom{n}{K}$ clauses is placed into the formula $\mathbf{\Phi}(n, dn)$ with probability $p_{n,d} = dn/(2^K\binom{n}{K})$, independently for each clause. Thus the expected number of clauses is dn. Conditional on generating N clauses, the new model is precisely $\mathbf{\Phi}(n, N)$. As before, by a simple concentration inequality, for every fixed $\epsilon > 0$ the probability that the actual number of clauses deviates from dn by more than ϵn is exponentially small in n. In particular, if instead we set $p_{n,d}$ to be $(1+\epsilon)dn/(2^K\binom{n}{K})$, the probability that the number of clauses in the new model is less than dn is exponentially small in n. Thus obtaining a probabilistic upper bound on the size of $\mathcal{IR}_{x_i}$ in the modified model implies the same bound on the original model $\mathbf{\Phi}(n, dn)$, with estimate difference at most exponential in n, which is subsumed by a smaller rate in (7). Thus we now switch to the new model, but for simplicity we assume that $p_{n,d} = dn/(2^K\binom{n}{K})$, dropping the $1 + \epsilon$ term. It will be easy to see that this does not impact the estimates. Also for simplicity we use $\mathbf{\Phi}(n, dn)$ to denote the modified model as well.

For every variable x we write $\mathcal{IR}_x = \cup_t \mathcal{IR}_{x,t}$, where we define $\mathcal{IR}_{x,t}$ to be the set of all variables y who are influenced by x through a path of length exactly t in graph $\mathbb{G}(\mathbf{\Phi}(n, dn))$. Note that the sets $\mathcal{IR}_{x,t}$ by this definition are not necessarily mutually disjoint since a variable x can influence another variable y via several paths

of different lengths. Thus $\mathcal{IR}_x = \cup_t \mathcal{IR}_{x,t}$. Let $I_{x,t} \triangleq |\mathcal{IR}_{x,t}|$. We split the analysis into two cases: $t \geq \ln n/(\ln\ln n)^{1-\xi/2} \triangleq \tau_n$ and $t < \tau_n$.

Suppose $t < \tau_n$. Then every node $y \in \mathcal{IR}_{x,t}$ is within distance

$$\begin{aligned} rt &\leq (\ln\ln n)^{1-\xi} \ln n/(\ln\ln n)^{1-\xi/2} \\ &= \ln n/(\ln\ln n)^{\xi/2} \\ &= o(\ln n) \end{aligned}$$

from x. Applying Lemma 3.4, we conclude that

$$\mathbb{P}\left(|\cup_{t\leq\tau_n} \mathcal{IR}_{x,t}| \geq n^{\epsilon}\right) \leq \exp\left(-n^{\epsilon/5}\right)$$

for all large enough n. Note that for this case we have not used the fact that the labels z_{y_i} of the influence path from x to $y \in \mathcal{IR}_{x,t}$ have to be ordered but have simply used a bound from Lemma 3.4 on the size of the neighborhoods around x.

Next we consider the case $t \geq \tau_n$. It is here that the ordering of labels z_{y_i} will be important. We claim that

$$\mathbb{E}\left[|\cup_{t\geq\tau_n} \mathcal{IR}_{x,t}|\right] \leq \exp\left(-(1/6)\ln n(\ln\ln n)^{\xi/3}\right) \tag{8}$$

for large enough n, which by Markov's inequality implies

$$\mathbb{P}\left(|\cup_{t\geq\tau_n} \mathcal{IR}_{x,t}| \geq 1\right) \leq \exp\left(-(1/6)\ln n(\ln\ln n)^{\xi/3}\right)$$

for large enough n. Combining the two bounds, we obtain

$$\begin{aligned} \mathbb{P}\left(|\cup_t \mathcal{IR}_{x,t}| \geq n^{\epsilon}\right) &\leq \exp\left(-n^{\epsilon/5}\right) + \exp\left(-(1/6)\ln n(\ln\ln n)^{\xi/3}\right) \\ &\leq \exp\left(-\ln n(\ln\ln n)^{\xi/4}\right) \end{aligned}$$

for all large enough n, and the proof of the proposition is complete by taking a union bound over the n choices of x.

We now establish (8). Consider an influence path of length t starting from x, namely a path $y_0 = x, y_1, \ldots, y_t$ such that $z_{y_0} > z_{y_1} > \cdots > z_{y_t}$ and such that y_i and y_{i+1} are connected by a path in $\mathbb{G}(\mathbf{\Phi}(n, dn))$ denoted by $y_0^i = y_i, y_1^i, \ldots, y_{r_i}^i = y_{i+1}$ with length $r_i \leq r$. This implies the existence of clauses denoted by $C_{i,j}$, $0 \leq i \leq t-1$, $0 \leq j \leq r_i - 1$, such that the clause $C_{i,j}$ contains both variables y_j^i and y_{j+1}^i for all $0 \leq i \leq t-1$, $1 \leq j \leq r_i$. Each such clause contains $K-2$ additional variables. We now fix any such path y_i, $0 \leq i \leq t$, we fix the corresponding "connecting" variables y_j^i, and we fix the clauses $C_{i,j}$. The probability that such a path exists in the graph $\mathbb{G}(\mathbf{\Phi}(n, dn))$, by the independence of choices of variables in the clauses, is

$$(2^K p_{n,d})^{\sum_{0\leq i\leq t-1} r_i} = \left(\frac{dn}{\binom{n}{K}}\right)^{\sum_{0\leq i\leq t-1} r_i},$$

where 2^K in front of $p_{n,d}$ accounts for the possibilities of negations in clauses. Given that this path exists in graph $\mathbb{G}(\mathbf{\Phi}(n, dn))$, the probability that it is also a path of influence is $1/t!$ by the independence of labels $\mathbf{z}$ from all other randomness in the model. The total number of such paths is crudely upper bounded by $((K-1)n)^{(K-1)\sum_{0\leq i\leq t-1} r_i - 1}$, where 1 is subtracted since the first variable $y^0 = y_0^0 = x$

is fixed, and factor $K-1$ in front of n accounts for $K-1$ choices for variables in a clause containing y_j^i serving as connectors with the next clause. We obtain the following upper bound:

$$\begin{aligned}\mathbb{E}[I_{x,t}] &\le (t!)^{-1}((K-1)n)^{(K-1)\sum_{0\le i\le t-1} r_i - 1}\left(\frac{dn}{\binom{n}{K}}\right)^{\sum_{0\le i\le t-1} r_i}\\ &\le K!\frac{(Kd)^{Krt}}{t!n} + O\left(\frac{(Kd)^{Krt}}{t!n^2}\right)\\ &= O\left(\frac{(Kd)^{Krt}}{t!n}\right), \qquad (9)\end{aligned}$$

where the constant hidden in $O(\cdot)$ may depend on K but not on r, t. Recall that $r \le (\ln\ln n)^{1-\xi}$. When $t \ge \tau_n$, we have $\ln t \ge \ln\ln n - (1-\xi/2)\ln^{(3)} n$, where $\ln^3$ denotes a three times iterated logarithm. Since $\xi > 0$, this implies $(Kr)\ln(Kd) \le (1/4)\ln t$ for large enough n. Then, using the Stirling's approximation which gives $t! \ge t^{\frac{t}{2}}$ for large enough t, we obtain that for all large enough n

$$\begin{aligned}\frac{(Kd)^{Krt}}{t!} &\le \exp(Krt\ln(Kd) - (t/2)\ln t)\\ &\le \exp\left(-(t/4)\ln t\right)\\ &\le \exp\left(-(1/5)\ln n(\ln\ln n)^{\xi/2}\right).\end{aligned}$$

Next observe that for every $\lambda > 0$ and $k_0 > 2\lambda$ we have $(2\lambda)^k/k! \le (2\lambda)^{k_0}/k_0!$ for all $k \ge k_0$. This implies

$$\begin{aligned}\sum_{k\ge k_0}\frac{\lambda^k}{k!} &\le \frac{\lambda^{k_0}}{k_0!}\sum_{k\ge k_0} 2^{-(k-k_0)}\\ &= 2\frac{\lambda^{k_0}}{k_0!}.\end{aligned}$$

We obtain that for large enough n,

$$\sum_{t\ge\tau_n}\frac{(Kd)^{Krt}}{t!} \le \exp\left(-(1/6)\ln n(\ln\ln n)^{\xi/2}\right),$$

where the factor 2, as well as the constant factor hidden in $O(\cdot)$ in (9), is consumed by lowering the factor 1/5 to 1/6. We then obtain

$$\begin{aligned}\mathbb{E}\left[\sum_{t\ge\tau_n} I_{x,t}\right] &= \sum_{t\ge\tau_n}\mathbb{E}[I_{x,t}]\\ &\le \exp\left(-(1/6)\ln n(\ln\ln n)^{\xi/3}\right)\end{aligned}$$

for all large enough n, and (8) is established. ∎

4. The overlap structure of nearly satisfying assignments. In this section we establish a certain property regarding overlaps of multiple assignments of the NAE-K-SAT problem. 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 $\mathbb{SAT}(\Phi, \ell)$ for the set of assignments violating at most $\ell \le dn$ clauses, and $\mathbb{SAT}(\Phi)$ for the set of satisfying assignments of a formula Φ, we have $\mathbb{P}(\mathbb{SAT}(\mathbf{\Phi}(n, dn)) \ne \emptyset) \to 1$ as $n \to \infty$ for every $d < d_s$.

The solution overlap property we consider in this section is with respect to the Hamming distance $\rho(\sigma^1, \sigma^2)$ between two assignments σ^1 and σ^2, denoted $\rho(\sigma^1, \sigma^2)$, which is the number of variables x_i with different assignments according to σ^1 and σ^2. In the prior literature the nontrivial overlap property of satisfying assignments was established by proving a certain clustering property, which says that the "satisfaction graph," the graph of satisfying assignments where two assignments are deemed adjacent if the Hamming distance between them is $o(n)$, has many connected components. A condition which in turn implies this simple notion is that for every pair of satisfying assignment σ^1 and σ^2 it is the case that $\rho(\sigma^1, \sigma^2)/n \notin (\beta - \eta, \beta)$ for some $\eta > 0$, and there are at least two solutions σ^1, σ^2 with $\rho(\sigma^1, \sigma^2)/n \ge \beta$. Note that this implies that any pair of satisfying assignments σ^1 and σ^3 with $\rho(\sigma^1, \sigma^3) > \beta n$ must be disconnected in the satisfaction graph, or else there will be a point σ^2 on the path between them with $\rho(\sigma^1, \sigma^2)/n \in (\beta - \eta, \beta)$.

Unfortunately, working purely with this notion, one gets such a clustering result only for very high densities d, specifically for d at least $d_s/2$. (We skip details since this fact is not needed for our main result.) To obtain a result for smaller density d we establish a more complicated nontrivial overlap property, which was inspired by the development in [RV14]. Roughly speaking, we show that there cannot be many assignments $\sigma^1, \dots, \sigma^m$ for some constant m which satisfy a certain minimum number of clauses such that all pairwise Hamming distances $\rho(\sigma^i, \sigma^j)$ fall between $(\beta - \eta)n$ and βn. We now give the formal definition.

Fix $\beta, \eta \in [0, 1]$, $\kappa \ge 0$, and a positive integer m. Given an NAE-K-SAT formula Φ, denote by $\mathbb{SAT}(\Phi; \beta, \eta, \kappa, m)$ the set of all m-tuples $(\sigma^1, \dots, \sigma^m)$ of assignments $\sigma^j : \{x_1, \dots, x_n\} \to \{0, 1\}$, $1 \le j \le m$, satisfying the following properties:

(a) Every σ^j, $1 \le j \le m$, is an assignment violating at most κn clauses. Namely, $\mathbb{SAT}(\Phi; \beta, \eta, \kappa, m) \subset \mathbb{SAT}^m(\Phi, \kappa n)$.

(b) For every j, k, $(\beta - \eta)n \le \rho(\sigma^j, \sigma^k) \le \beta n$.

In our application we will choose η to be much smaller than β. In this case the pairwise distances $\rho(\sigma^j, \sigma^k)$ are nearly βn. Thus we may think of such an m-tuple as a set of m equidistant points in the Hamming cube $\{0, 1\}^n$ with pairwise distances nearly βn.

We now state the main result of this section.

THEOREM 4.1. *Fix arbitrary $0 < \epsilon < 1$, and let $\beta = \frac{\ln K}{K}$, $\eta = \left(\frac{\ln K}{K}\right)^2$, $\kappa = (2\ln(2))^{-1}\epsilon^3 \ln^2 K/K^2$, and $m = \lceil \frac{\epsilon^2 K}{\ln K} \rceil$. Then there exist $K_0 = K_0(\epsilon)$ and $\delta > 0$ such that for all $K \ge K_0$ and $d \ge (1 + \epsilon)2^{K-1}\ln^2 K/K$ the following holds:*

$$\lim_{n\to\infty} n^{-1} \ln \mathbb{P}\left(\mathbb{SAT}\left(\mathbf{\Phi}(n, dn), \beta, \eta, \kappa, m\right) \ne \emptyset\right) \le -\delta.$$

Intuitively Theorem 4.1 states that for certain choices of β, η, κ, and m which depend on K only, when d crosses the threshold $\approx (d_s/K)\ln^2 K$, the probability of finding m equidistance assignments which violate at most $\approx \kappa n$ clauses is at most $e^{-\delta n}$ for large enough K and n.

Proof. The proof is based on the application of the first moment argument. We consider the expected number of m-tuples satisfying the conditions (a)–(b) and show

that this expectation converges to zero exponentially fast as $n \to \infty$. Applying Markov's inequality, the result then will follow.

We begin by computing asymptotically the number of m-tuples $\sigma^1, \ldots, \sigma^m$ satisfying condition (b) only. We have 2^n choices for σ^1. For any fixed choice of σ^1 and any fixed $j = 2, \ldots, m$ the number of choices for σ^j is

$$\sum_{n(\beta-\eta) \le r \le n\beta} \binom{n}{r},$$

by considering all the subsets of variables $x_1, \ldots, x_n$ where σ^1 and σ^j disagree. Since this applies for every j, we obtain the following upper bound on the number of m-tuples satisfying (b):

$$2^n \left(\sum_{n(\beta-\eta) \le r \le n\beta} \binom{n}{r} \right)^{m-1}.$$

This bound appears to be loose, since it ignores the constraints on $\rho(\sigma^j, \sigma^k)$ for $j, k \ge 2$. Nevertheless, it suffices for our purposes. We now obtain an asymptotic upper bound on this expression in terms of ϵ, K, and n.

Using Stirling's approximation and since the function $-x \ln x$ is increasing in the range $x < e^{-1}$ and decreasing in the range $x > e^{-1}$, the expression is at most

$$\exp\left(n \ln 2 - nm\beta \ln \beta - nm(1 - \beta + \eta) \ln(1 - \beta + \eta) + o(n)\right). \tag{10}$$

Here we use $\beta = \ln K/K < e^{-1}$ and $1 - \beta - \eta = 1 - \ln K/K > e^{-1}$ for sufficiently large K. Further, the same asymptotics gives $-\ln \beta = \ln K + O_K(\ln \ln K)$, implying

$$\begin{aligned} -m\beta \ln \beta &= m\left(\beta \ln K + \beta O_K(\ln \ln K)\right) \\ &= \epsilon^2 \ln K + O_K(\ln \ln K). \end{aligned}$$

Next, we have for sufficiently large K

$$\begin{aligned} -m(1 - \beta + \eta) \ln(1 - \beta + \eta)) &\le m((\ln K/K) + o_K(\ln K/K)) \\ &\le \epsilon^2 + o_K(1). \end{aligned}$$

We conclude that for sufficiently large K the term (10) is at most

$$\exp(n\epsilon^2 \ln K + nO_K(\ln \ln K)). \tag{11}$$

We now compute an upper bound on the probability that a given m-tuple $\sigma^1, \ldots, \sigma^m$ satisfying (b) consists of assignments violating at most κn clauses. Should this be the case, then the total number of violated clauses is at most $m\kappa n$, and thus there exist at least $dn - m\kappa n = rn$ clauses satisfied by all of the assignments $\sigma^1, \ldots, \sigma^m$, where $r \triangleq d - m\kappa$. We fix any set of clauses with cardinality rn, which without loss of generality we assume to be $C_1, \ldots, C_{rn}$, and obtain an upper bound on the probability that each $\sigma^j, 1 \le j \le m$, satisfies every clause $C_1, \ldots, C_{rn}$. Then we will take the union bound on the all subsets of $C_1, \ldots, C_{dn}$ of cardinality rn.

Let C be a clause generated uniformly at random from the space of all clauses (a generic element of the formula $\mathbf{\Phi}(n, dn)$). Applying the truncated exclusion-inclusion

principle, the probability that C is satisfied by every assignment $\sigma^1, \ldots, \sigma^m$ is

$$\begin{aligned}\mathbb{P}(C \text{ satisfied by } \sigma^j\ \forall j = 1, \ldots, m) &= 1 - \mathbb{P}(\exists j :\ C \text{ is not satisfied by } \sigma^j,\ 1 \le j \le m) \\ &\le 1 - \sum_{1 \le j \le m} \mathbb{P}(C \text{ is not satisfied by } \sigma^j) \\ &\quad + \sum_{1 \le j_1 < j_2 \le m} \mathbb{P}(C \text{ is not satisfied by } \sigma^{j_1}, \sigma^{j_2}).\end{aligned}$$

Now $\mathbb{P}(C \text{ is not satisfied by } \sigma^j) = 2^{-K+1}$. Also for every two assignments σ^1 and σ^2 which disagree in $n_0 \le n$ variables

$$\mathbb{P}(C \text{ is not satisfied by } \sigma^1, \sigma^2) = 2^{-K+1}\left(\left(\frac{n_0}{n}\right)^K + \left(1 - \frac{n_0}{n}\right)^K\right).$$

We conclude that, for every m-tuple $\sigma_1, \ldots, \sigma_m$ satisfying (b), the probability that this m-tuple satisfies clauses $C_1, \ldots, C_{rn}$ is at most

$$\begin{aligned}&\left(1 - m2^{-K+1} + (m(m-1)/2)2^{-K+1}(\beta^K + (1 - \beta + \eta)^K)\right)^{rn} \\ &\le \left(1 - \epsilon^2 K(\ln K)^{-1}2^{-K+1} + \epsilon^4 K^2(\ln K)^{-2}2^{-K+2}(K^{-1} + o_K(K^{-1}))\right)^{rn}.\end{aligned}$$

Here we used the fact that for $\beta = \ln K/K$ and $\eta = (\ln K/K)^2$ we have

$$\ln \beta^K + (1 - \beta + \eta)^K = K^{-1} + o_K(K^{-1}).$$

The upper bound then simplifies to

$$\left(1 - \epsilon^2 K(\ln K)^{-1}2^{-K+1} + o_K(K(\ln K)^{-1}2^{-K})\right)^{rn},$$

which using $r = d - m\kappa$ and applying the lower bound $d \ge (1 + \epsilon)(2^{K-1}/K)\ln^2 K$, leads to a bound

$$\exp\left(-n(1 + \epsilon)\epsilon^2 \ln K + no_K(\ln K)\right). \tag{12}$$

On the other hand, the number of ways of choosing rn out of dn clauses using Stirlings' approximation is

$$\binom{dn}{m\kappa n} = \exp\left(-dn(m\kappa/d)\ln(m\kappa/d) - dn(1 - m\kappa/d)\ln(1 - m\kappa/d) + o(n)\right).$$

We now analyze the exponent. Since $1/(m\kappa) = O_K(K/\ln K)$, then

$$\begin{aligned}-\ln(m\kappa/d) &= \ln(d) - \ln(m\kappa) \\ &= K\ln 2 + o_K(K).\end{aligned}$$

Thus

$$-d(m\kappa/d)\ln(m\kappa/d) = m\kappa K\ln(2)(1 + o_K(1)),$$

which by our choice of κ is $(1/2)\epsilon^3 \ln K + o_K(\ln K)$. For the second term in the exponent, using a first order Taylor expansion,

$$\begin{aligned}-d(1 - m\kappa/d)\ln(1 - m\kappa/d) &= dm\kappa/d \\ &= m\kappa \\ &= o_K(\ln K).\end{aligned}$$

Applying our bounds for (12), we obtain the following upper bound on the probability that there exist at least rn clauses satisfied by all of the assignments $\sigma^1, \dots, \sigma^m$:

$$\begin{aligned}&\exp\left(-n(1+\epsilon)\epsilon^2 \ln K + (1/2)n\epsilon^3 \ln K + no_K(\ln K)\right)\\ &\quad = \exp\left(-n\epsilon^2 \ln K - (1/2)n\epsilon^3 \ln K + no_K(\ln K)\right).\end{aligned}$$

Now combining this with (11), we conclude that the expected number of m-tuples satisfying conditions (a) and (b) is at most

$$\exp(-n(1/2)\epsilon^3 \ln K + no_K(\ln K)).$$

The proof of the theorem is complete by applying Markov's inequality. ∎

5. Proof of Theorem 2.4. The main result of this section states that if a τ-decimation algorithm works well on random instances of NAE-K-SAT, then it can be run several times to produce several nearly satisfying assignments, and in particular such that their overlaps (Hamming distances) satisfy properties (a) and (b) described in the previous sections with parameters β, η, κ, and m given in Theorem 4.1. Since such overlaps are "forbidden" by this theorem, we will obtain a contradiction. We state our main proposition below and show how Theorem 2.4 follows almost immediately. The rest of this section is devoted to the proof of the proposition.

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, ordering given by $\mathbf{Z}$, and using $\mathbf{U}$ to determine the rounding of the probabilities given by τ. Recall that $\rho(\sigma^1, \sigma^2)$ denotes the Hamming distance between assignments σ^1 and σ^2. Let κ again be defined by (1). Let α_n denote the probability that the τ-decimation algorithm finds an assignment in a random formula $\mathbf{\Phi}(n, dn)$ violating at κn clauses. Namely, $\alpha_n = \mathbb{P}\left(\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{U}} \in \mathbb{SAT}(\mathbf{\Phi}(n,dn), \kappa n)\right)$.

PROPOSITION 5.1. *Suppose* $\alpha_n > \exp\left(-\ln n(\ln\ln)^{\xi/6}\right)$ *for all large enough* n. *Then for every* $0 < \eta < \beta$ *such that* $[\beta - \eta, \beta] \subset [0, 1/2]$ *and every positive integer* m, K, *and* d,

$$\mathbb{P}_{\mathbf{\Phi}(n,dn)}\left(\mathbb{SAT}(\mathbf{\Phi}(n,dn); \beta, \eta, \kappa, m) \neq \emptyset\right) \geq \exp\left(-\ln n\,(\ln\ln)^{\xi/5}\right)$$

for all sufficiently large n.

Proof of Theorem 2.4. The result follows immediately from Theorem 4.1 and Proposition 5.1 by setting β, η, and m exactly as in Theorem 4.1 and noting that $[\beta-\eta, \beta] \subset [0, 1/2]$ is satisfied for sufficiently large K and, on the other hand, observing that

$$\exp\left(-\ln n\,(\ln\ln)^{\xi/5}\right) > \exp(-\delta/2n)$$

for all large enough n, where δ is as in Theorem 4.1. ∎

5.1. Proof of Proposition 5.1.

Proof of Proposition 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 m independent vectors $\mathbf{U}^0, \dots, \mathbf{U}^{m-1}$ which can be used to generate assignments. By definition we have

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

for $j = 0, \ldots, m-1$. We now construct a sequence of vectors $\mathbf{V}^{t,j}$, $0 \le t \le n$, $0 \le j \le m-1$, where for each $j = 1, \ldots, m-1$ the sequence $\mathbf{V}^{t,j}$ will interpolate between vectors $\mathbf{U}^0$ and $\mathbf{U}^j$. Specifically, let $\mathbf{V}^{t,j} = (V_1^{t,j}, \ldots, V_n^{t,j})$, where $V_i^{t,j} = U_i^j$, $i \le t$, and $V_i^{t,j} = U_i^0$, $t < i \le n$. Note that for every $t = 0, 1, \ldots, n$, $\mathbf{V}^{t,j}$ is a vector of i.i.d. random variables with a distribution uniform in $[0,1]$. Furthermore, $\mathbf{V}^{0,j} = \mathbf{U}^0$, $\mathbf{V}^{t,0} = \mathbf{U}^0$, and $\mathbf{V}^{n,j} = \mathbf{U}^j$. 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 the assignment of x_t by the τ-decimation algorithm. Observe that, given any realizations $\mathbf{u}^j$, $0 \le j \le m-1$, of vectors $\mathbf{U}^j$, and the corresponding realizations $\mathbf{v}^{t,j}$ of $\mathbf{V}^{t,j}$, we have

$$\rho(\sigma_{\Phi,\mathbf{z},\mathbf{v}^{t+1,j}}, \sigma_{\Phi,\mathbf{z},\mathbf{v}^{t,j}}) \le |\mathcal{IR}_{x_{t+1}}|, \qquad 0 \le t \le n-1, \tag{13}$$

since $\mathbf{v}^{t,j}$ and $\mathbf{v}^{t+1,j}$ differ only in one coordinate $t+1$, and by Proposition 3.3, changing the value of u_{t+1} impacts only the decisions for variables in $\mathcal{IR}_{x_{t+1}}$.

LEMMA 5.2. *For all large enough n the following holds:*

$$\mathbb{P}\left(\forall\ 0 \le j_1 \ne j_2 \le m-1,\ \rho(\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{V}^{T,j_1}}, \sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{V}^{T,j_2}}) \in [(\beta-\eta)n, \beta n]\right)$$
$$\ge 1 - \exp\left(-\ln n (\ln\ln n)^{\xi/5}\right). \tag{14}$$

Thus, per Lemma 5.2, the sequence of assignments $\sigma^j \triangleq \sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{V}^{T,j}}$, $0 \le j \le m-1$, satisfies property (b) of the definition of $\mathbb{SAT}(\Phi; \beta, \kappa, \eta, m)$ with probability at least $1 - \exp\left(-\ln n (\ln\ln n)^{\xi/5}\right)$ for large enough n.

Proof. We now consider a realization Φ of a formula $\mathbf{\Phi}(n, dn)$ and realization $\mathbf{z}$ of the order $\mathbf{Z}$. Φ and $\mathbf{z}$ uniquely determine sets $\mathcal{IR}_{x_i}, 1 \le i \le n$. Let $\mathcal{E}_n$ denote the event (the set of Φ and $\mathbf{z}$) that $\max_{1 \le i \le n} |\mathcal{IR}_{x_i}| \le n^{1/6}$. By Proposition 3.5 we have

$$\mathbb{P}(\mathcal{E}_n) \ge 1 - \exp\left(-\ln n (\ln\ln n)^{\xi/4}\right) \tag{15}$$

for large enough n. Here the choice of $1/6$ is somewhat arbitrary, and in fact any value less than $1/5$ is fine by Proposition 3.5. We assume without loss of generality that n is large enough so that $n^{1/6} < (\beta - \eta)n$.

We first suppose that Φ and $\mathbf{z}$ are realizations such that the event $\mathcal{E}_n$ takes place. We have by property (4) of Lemma 3.1 that, for every Φ and $\mathbf{z}$,

$$\mathbb{E}[\rho(\sigma_{\Phi,\mathbf{z},\mathbf{U}^0}, \sigma_{\Phi,\mathbf{z},\mathbf{U}^j})] = n/2$$

for each $j = 1, \ldots, m-1$. Here the randomness is with respect to $\mathbf{V}^{t,j}$, as Φ and $\mathbf{z}$ are fixed. Then, we can find $t_0 = t_0(\Phi, \mathbf{z})$ such that

$$\mathbb{E}[\rho(\sigma_{\Phi,\mathbf{z},\mathbf{U}^0}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j}})|\Phi, \mathbf{z}] \in \left[(\beta - \eta/2)n, (\beta-\eta/2)n + n^{1/6}\right]$$

for all $j = 1, \ldots, m-1$, as by (13) the increments $\rho(\sigma_{\Phi,\mathbf{z},\mathbf{V}^{t+1}}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^{t,j}})$ are bounded by $n^{1/6}$ with probability one with respect to the randomness of $\mathbf{V}^{t,j}$.

Note that t_0 does not depend on j since $\mathbf{V}^{t,j}$ are identically distributed for $1 \le j \le m-1$. Furthermore, since $\mathbf{U}^0$ and $\mathbf{U}^j$ are identical in distribution, we also have for every $0 \le j_1 < j_2 \le m-1$

$$\mathbb{E}[\rho(\sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j_1}}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j_2}})] \in \left[(\beta - \eta/2)n, (\beta-\eta/2)n + n^{1/6}\right].$$

We now fix $j_1 \neq j_2$ and argue that in fact $\rho(\sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j_1}}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j_2}})$ is concentrated around its mean as $n \to \infty$. The distance is a function of $n+t_0$ i.i.d. random variables $U_1^{j_1}, \ldots, U_{t_0}^{j_1}; U_1^{j_2}, \ldots, U_{t_0}^{j_2}; U_{t_0+1}^0, \ldots, U_n^0$. Further, changing any one of these $n+t_0$ random variables changes the distance ρ by at most $2n^{1/6}$, again by Proposition 3.3 and by our assumption that Φ and $\mathbf{z}$ are realizations such that the event $\mathcal{E}_n$ holds. Applying Azuma's inequality,

$$\begin{aligned}
&\mathbb{P}\left(\left|\rho(\sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j_1}}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j_2}}) - (\beta-\eta/2)n\right| \geq \frac{\eta}{4}n\right) \\
&\quad \leq 2\exp\left(-\frac{(\frac{\eta}{4}n - n^{\frac{1}{6}})^2}{4(n+t_0)n^{\frac{1}{6}}}\right) \\
&\quad = \exp\big(-\delta n^{5/6} + o(n^{\frac{5}{6}})\big)
\end{aligned}$$

for some constant $\delta > 0$, and the concentration is established. The event

$$\left|\rho(\sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j_1}}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j_2}}) - (\beta-\eta/2)n\right| < \frac{\eta}{4}n$$

implies the event

$$\rho(\sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j_1}}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j_2}}) \in [(\beta-\eta)n, \beta n].$$

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

$$\mathbb{P}\left(\rho(\sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j_1}}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j_2}}) \in [(\beta-\eta n), \beta n]\right) \geq 1 - \exp\big(-\delta n^{5/6} + o(n^{\frac{5}{6}})\big) \tag{16}$$

for all n satisfying $n^{1/6} < (\beta-\eta)n$. Since m does not depend on n, we obtain by the union bound

$$\begin{aligned}
\mathbb{P}\Big(\forall\ 0 \leq j_1 \neq j_2 \leq m-1,\ &\rho(\sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j_1}}, \sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j_2}}) \\
&\in [(\beta-\eta)n, \beta n]\Big) \geq 1 - \exp(-\delta n^{2/3})
\end{aligned} \tag{17}$$

for all large enough n, where again the choice $2/3$ was arbitrary as long as it is smaller than $5/6$. 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. Let now $T = t_0(\mathbf{\Phi}(n, dn), \mathbf{Z})$ be a random variable thus defined. This way we have assignments $\sigma_{\Phi,\mathbf{z},\mathbf{V}^{T,j}}$, $0 \leq j \leq m-1$, defined for all realizations of Φ and $\mathbf{z}$, in particular whether the event $\mathcal{E}_n$ takes place or not. Since the former is the high probability event, we conclude from above that

$$\begin{aligned}
&\mathbb{P}\Big(\forall\ 0 \leq j_1 \neq j_2 \leq m-1,\ \rho(\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{V}^{T,j_1}}, \sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{V}^{T,j_2}}) \in [(\beta-\eta)n, \beta n]\Big) \\
&\quad \geq 1 - \exp(-\delta n^{2/3}) - \exp\big(-\ln n(\ln\ln n)^{\xi/4}\big) \\
&\quad \geq 1 - \exp\big(-\ln n(\ln\ln n)^{\xi/5}\big)
\end{aligned}$$

for all large enough n, and the bound (14) is established. ∎

Our next goal is to show that the assignments σ^j, $0 \leq j \leq m-1$, above are also κn-satisfying formula $\mathbf{\Phi}(n, dn)$ with probability at least α_n^m, for large enough n. Namely, we have the following result.

LEMMA 5.3. *For all large enough n,*

$$\mathbb{P}(\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{V}^{T,j}} \in \mathbb{SAT}(\mathbf{\Phi}(n, dn), \kappa n),\ 0 \leq j \leq m-1) \geq \alpha_n^m. \tag{18}$$

Proof. Observe that $\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{V}^{T,j}}$ have identical distribution for all j. Furthermore, each of them individually is distributed as $\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{U}^j}$, $0 \le j \le m-1$, since the random variable T only affects the indices i for which we switch from U_i^0 vs U_i^j, and since each vector $\mathbf{U}^j$ is an i.i.d. vector of random variables. Therefore,

$$\mathbb{P}(\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{V}^{T,j}} \in \mathbb{SAT}(\mathbf{\Phi}(n,dn))) = \alpha_n$$

for each j. Suppose $\Phi, \mathbf{z}$ are such that the event $\mathcal{E}_n$ takes place, and fix the corresponding deterministic value $t_0 = t_0(\Phi, \mathbf{z})$. In the derivation below we use notation $\mathbb{P}_Z$ to indicate probability with respect random variable Z. We have

$$\begin{aligned}
&\mathbb{P}_{\mathbf{U}^0,\dots,\mathbf{U}^{m-1}}(\sigma_{\Phi,\mathbf{z},\mathbf{V}^{T,j}} \in \mathbb{SAT}(\Phi),\ 0 \le j \le m-1) \\
&= \mathbb{E}_{\mathbf{U}^0,\dots,\mathbf{U}^{m-1}}[\mathbf{1}(\sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j}} \in \mathbb{SAT}(\Phi),\ 0 \le j \le m-1)] \\
&= \mathbb{E}_{U^0_{t_0+1},\dots,U^0_n}\big[\mathbb{E}_{U_i^j,1\le i\le t_0,1\le j\le m-1}[\mathbf{1}(\sigma_{\Phi,\mathbf{z},\mathbf{V}^{t_0,j}} \in \mathbb{SAT}(\Phi)), \\
&\qquad\qquad 0 \le j \le m-1 \mid U^0_{t_0+1},\dots,U^0_n]\big]. \qquad (19)
\end{aligned}$$

Since $\mathbf{U}^j$ are independent vectors of i.i.d. random variables, the last expression equals

$$\mathbb{E}_{U^0_{t_0+1},\dots,U^0_n}\left[\mathbb{E}^m_{U^0_1,\dots,U^0_{t_0}}\left[\mathbf{1}(\sigma_{\Phi,\mathbf{z},\mathbf{U}^0} \in \mathbb{SAT}(\Phi)) \mid U^0_{t_0+1},\dots,U^0_n\right]\right]. \qquad (20)$$

Applying Jensen's inequality and the convexity of the polynomial function t^m on $t \in [0,\infty)$ for all positive integers m, we obtain

$$\begin{aligned}
&\mathbb{E}_{U^0_{t_0+1},\dots,U^0_n}\left[\mathbb{E}^m_{U^0_1,\dots,U^0_{t_0}}\left[\mathbf{1}(\sigma_{\Phi,\mathbf{z},\mathbf{U}^0} \in \mathbb{SAT}(\Phi)) \mid U^0_{t_0+1},\dots,U^0_n\right]\right] \\
&\ge \mathbb{E}^m_{U^0_{t_0+1},\dots,U^0_n}\left[\mathbb{E}_{U^0_1,\dots,U^0_{t_0}}\left[\mathbf{1}(\sigma_{\Phi,\mathbf{z},\mathbf{U}^0} \in \mathbb{SAT}(\Phi)) \mid U^0_{t_0+1},\dots,U^0_n\right]\right] \\
&= \mathbb{E}^m_{\mathbf{U}^0}\left[\mathbf{1}(\sigma_{\Phi,\mathbf{z},\mathbf{U}^0} \in \mathbb{SAT}(\Phi))\right] \\
&= \mathbb{P}^m_{\mathbf{U}^0}(\sigma_{\Phi,\mathbf{z},\mathbf{U}^0} \in \mathbb{SAT}(\Phi)). \qquad (21)
\end{aligned}$$

Suppose now that Φ and $\mathbf{z}$ are such that the event $\mathcal{E}_n$ does not take place. Then $\sigma_{\Phi,\mathbf{z},\mathbf{V}^{T,j}} = \sigma_{\Phi,\mathbf{z},\mathbf{U}^0}$, implying

$$\begin{aligned}
\mathbb{P}_{\mathbf{U}^0,\dots,\mathbf{U}^{m-1}}(\sigma_{\Phi,\mathbf{z},\mathbf{V}^{T,j}} \in \mathbb{SAT}(\Phi),\ 0 \le j \le m-1) &= \mathbb{P}_{\mathbf{U}^0}(\sigma_{\Phi,\mathbf{z},\mathbf{U}^0} \in \mathbb{SAT}(\Phi)) \\
&\ge \mathbb{P}^m_{\mathbf{U}^0}(\sigma_{\Phi,\mathbf{z},\mathbf{U}^0} \in \mathbb{SAT}(\Phi)).
\end{aligned}$$

Combining this with (21), we conclude that for every $\Phi, \mathbf{z}$ we have

$$\mathbb{P}_{\mathbf{U}^0,\dots,\mathbf{U}^{m-1}}(\sigma_{\Phi,\mathbf{z},\mathbf{V}^{T,j}} \in \mathbb{SAT}(\Phi),\ 0 \le j \le m-1) \ge \mathbb{P}^m_{\mathbf{U}^0}(\sigma_{\Phi,\mathbf{z},\mathbf{U}^0} \in \mathbb{SAT}(\Phi)).$$

Since $\mathbb{P}_{\mathbf{U}^0}(\sigma_{\Phi,\mathbf{z},\mathbf{U}^0} \in \mathbb{SAT}(\Phi)) = \alpha_n$, then integrating over $\mathbf{\Phi}(n,dn)$ and $\mathbf{Z}$, we obtain

$$\begin{aligned}
&\mathbb{P}(\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{V}^{T,j}} \in \mathbb{SAT}(\mathbf{\Phi}(n,dn)),\ 0 \le j \le m-1) \\
&\qquad \ge \mathbb{P}(\sigma_{\mathbf{\Phi}(n,dn),\mathbf{Z},\mathbf{U}^0} \in \mathbb{SAT}(\mathbf{\Phi}(n,dn)) \\
&\qquad = \alpha_n^m,
\end{aligned}$$

and (18) is established. This concludes the proof of Lemma 5.3. ∎

Combining Lemmas 5.3 and 5.2, the set $\mathbb{SAT}(\mathbf{\Phi}(n,dn);\beta,\eta,\kappa,m)$ is nonempty with probability at least

$$\begin{aligned}
1 - \exp\left(-\ln n(\ln\ln n)^{\xi/5}\right) - (1-\alpha_n^m) &= \alpha_n^m - \exp\left(-\ln n(\ln\ln n)^{\xi/5}\right) \\
&> \exp\left(-m\ln n(\ln\ln n)^{\xi/6}\right) - \exp\left(-\ln n(\ln\ln n)^{\xi/5}\right) \\
&\ge \exp\left(-\ln n(\ln\ln n)^{\xi/5}\right)
\end{aligned}$$

for all sufficiently large n, and thus the proof of Proposition 5.1 is complete. ∎

Appendix A. A detailed description of the SP-guided algorithm and its balancing property. The setup is similar to that 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 simply formally present the SP algorithm and SP-guided decimation algorithm, following closely [MM09] with the appropriate adjustment from the K-SAT problem to the NAE-K-SAT problem. We will convince ourselves that SP-guided decimation algorithm is again a special case of a balanced τ-decimation algorithm. We will then be able to conclude that the 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 notation are very similar to those described in [MM09]. Specifically, iterations (22)–(26) below correspond to iterations (20.17)–(20.20) in that book. Consider an arbitrary reduced or nonreduced 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 bipartite factor graph representation) is associated with five random variables, $Q^t_{x,C,U}$, $Q^t_{x,C,S}$, $Q^t_{x,C,*}$, $Q^t_{C,x,S}$, and $Q^t_{C,x,U}$. Here is the interpretation of these variables. Each of them is a message sent from a 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 a value which does not (does) satisfy C. $Q^t_{x,C,*}$ represents that none of these forcings takes place. $Q^t_{C,x,S}$ is interpreted as a probability computed at iteration t that all variables $y \in C$ other than x do not satisfy C, and thus that the only hope of satisfying C is for x to do so. Similarly, $Q^t_{C,x,U}$ is the probability that all variables y in C other than x do satisfy C and thus that 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 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 $Q^0_{x,C,U} + Q^0_{x,C,S} + Q^0_{x,C,*}$. Similarly, variables $Q^0_{C,x,S}$ and $Q^0_{C,x,U}$ are normalized to sum to one.

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 that $Q^t_{x,C,U}$, $Q^t_{x,C,S}$, $Q^t_{x,C,*}$, $Q^t_{C,x,S}$, and $Q^t_{C,x,U}$ are defined. Define the random variable $Q^{t+1}_{x,C,S}$ and $Q^{t+1}_{x,C,U}$ as follows. Suppose C is unsigned in

Φ. Then

$$Q^{t+1}_{C,x,S} = \prod_{y \in C \setminus x} Q^t_{y,C,U} \tag{22}$$

and

$$Q^{t+1}_{C,x,U} = \prod_{y \in C \setminus x} Q^t_{x,C,S}. \tag{23}$$

Here $C \setminus x$ is the set of variables in clause C other than x. The interpretation for these 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 were forced not to satisfy C at previous iterations 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 relevant only for the 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 $Q^{t+1}_{C,x,S} = 0$ and

$$Q^{t+1}_{C,x,U} = \prod_{y \in C \setminus x} Q^t_{x,C,S}. \tag{24}$$

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 to violate it is when all other variables y are forced to satisfy C. Again this is relevant only for the NAE-K-SAT problem. Similarly, if C is signed $-$, then $Q^{t+1}_{C,x,U} = 0$ and

$$Q^{t+1}_{C,x,S} = \prod_{y \in C \setminus x} Q^t_{x,C,U}. \tag{25}$$

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:

$$\begin{aligned} R^{t+1}_{x,C,S} = &\prod_{D \in \mathcal{U}_{x,C}} (1 - Q^t_{D,x,S}) \prod_{D \in \mathcal{S}_{x,C}} (1 - Q^t_{D,x,U}) \\ &- \prod_{D \in \mathcal{U}_{x,C}} (1 - Q^t_{D,x,*}) \prod_{D \in \mathcal{S}_{x,C}} (1 - Q^t_{D,x,*}), \end{aligned} \tag{26}$$

which is interpreted as follows. The first term on the right-hand side of the expression above is interpreted as the probability that none of the clauses D in $\mathcal{U}_{x,C}$ forces x to take a value which satisfies D and therefore violates C (since otherwise a contradiction would be reached) and none of the clauses D in $\mathcal{S}_{x,C}$ forces x to take value which violates D and therefore violates C (since otherwise a contradiction would be reached). The second term on the right-hand side is interpreted as the probability variable that x is not forced to take any particular value by clauses it belongs to other than C. The difference of the two terms is precisely the probability that x is forced to take a value satisfying C and is not forced to take a value contradicting this choice.

Similarly, define

$$\begin{aligned} R^{t+1}_{x,C,U} = &\prod_{D \in \mathcal{U}_{x,C}} (1 - Q^t_{D,x,U}) \prod_{D \in \mathcal{S}_{x,C}} (1 - Q^t_{D,x,S}) \\ &- \prod_{D \in \mathcal{U}_{x,C}} (1 - Q^t_{D,x,*}) \prod_{D \in \mathcal{S}_{x,C}} (1 - Q^t_{D,x,*}). \end{aligned} \tag{27}$$

The interpretation for $R^{t+1}_{x,C,U}$ is similar: it is the probability that x is forced to take a value violating C and is not forced to take a value satisfying C. Next, define

$$R^{t+1}_{x,C,*} = \prod_{D \in \mathcal{S}_{x,C} \cup \mathcal{U}_{x,C}} (1 - Q^t_{D,x,S} - Q^t_{D,x,U}). \tag{28}$$

$R^{t+1}_{x,C,*}$ is interpreted as the probability that x is not forced in either way by clauses other than C. Finally, we let $Q^{t+1}_{x,C,S}$, $Q^{t+1}_{x,C,U}$, and $Q^{t+1}_{x,C,*}$ 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 (22)–(26) are conducted for some number of steps $t = 0, 1, \ldots, r$. Next variables $W_x(1)$, $W_x(0)$, 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

$$W_x(1) = \prod_{D \in \mathcal{U}_x} (1 - Q^t_{D,x,S}) \prod_{D \in \mathcal{S}_x} (1 - Q^t_{D,x,U}) - \prod_{D \in \mathcal{U}_x} (1 - Q^t_{D,x,*}) \prod_{D \in \mathcal{S}_x} (1 - Q^t_{D,x,*}). \tag{29}$$

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

$$W_x(0) = \prod_{D \in \mathcal{S}_x} (1 - Q^t_{D,x,S}) \prod_{D \in \mathcal{U}_x} (1 - Q^t_{D,x,U}) - \prod_{D \in \mathcal{S}_x} (1 - Q^t_{D,x,*}) \prod_{D \in \mathcal{U}_x} (1 - Q^t_{D,x,*}), \tag{30}$$

with a similar interpretation. Then set

$$W_x(*) = \prod_{D \in \mathcal{S}_x \cup \mathcal{U}_x} (1 - Q^r_{D,x,S} - Q^r_{D,x,U}), \tag{31}$$

which is interpreted as the probability (after normalization) that x is not 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 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} is computed, and x_2 is set to 1 if $W_{x_2}(1) > W_{x_2}(0)$ and 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 a special case of the τ-decimation algorithm, where the 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.

We now turn to the proof of Observation 2.6.

Proof of Observation 2.6. Recall that at the iteration $t = 0$, the variables Q^t are chosen independently uniformly at random from $[0, 1]$, normalized appropriately. The main idea of the proof is to use the symmetry of the uniform distribution. Given a

formula Φ, we claim that if we initialize random variables Q^r with variables $Q^0_{x,C,U}$ and $Q^0_{x,C,S}$ swapped, variables $Q^0_{C,x,S}$ and $Q^0_{C,x,U}$ swapped, variables $Q^0_{x,C,*}$ left intact, and apply it to formula $\bar{\Phi}$ instead of Φ, we obtain values $W_x(0)$, $W_x(1)$, and $W_x(*)$ such that under this initialization $W_x(1) > W_x(0)$ holds if and only if $W_x(0) < W_x(1)$ under the original initialization for the original formula Φ. The claim of the proposition then follows.

We now establish the claim by a simple inductive reasoning. As suggested above, given $Q^0_{x,C,U}$, $Q^0_{x,C,S}$, $Q^0_{x,C,*}$, $Q^0_{C,x,S}$, and $Q^0_{C,x,U}$ (after normalization for concreteness), define

$$\begin{aligned} P^0_{x,C,U} &= Q^0_{x,C,S}, \\ P^0_{x,C,S} &= Q^0_{x,C,U}, \\ P^0_{x,C,*} &= Q^0_{x,C,*}, \\ P^0_{C,x,S} &= Q^0_{C,x,U}, \\ P^0_{C,x,U} &= Q^0_{C,x,S}. \end{aligned} \tag{32}$$

Then define variables $P^t_{x,C,U}$, $P^t_{x,C,S}$, $P^t_{x,C,*}$, $P^t_{C,x,S}$, and $P^t_{C,x,U}$ with respect to the formula $\bar{\Phi}$ similarly to the way variables $Q^t_{x,C,U}$, $Q^t_{x,C,S}$, $Q^t_{x,C,*}$, $Q^t_{C,x,S}$, and $Q^t_{C,x,U}$ are defined with respect to the formula Φ. We now prove by induction that the identities (32) hold for general t and not just when $t = 0$. The base of the induction is given by (32). Assume the claim holds for $t' \le t-1$. Consider any unsigned clause C in $\bar{\Phi}$. Then this clause is unsigned in Φ as well. Applying (22) and (23) and the inductive assumption, we conclude that the claim holds for $P^t_{C,x,S}$ and $P^t_{C,x,U}$ as well. Similarly, if a clause C is signed $+$ in $\bar{\Phi}$, then it is signed $-$ in Φ. Applying identities (24) and (25), the claim holds for $P^t_{C,x,S}$ and $P^t_{C,x,U}$ as well. The case when C is signed $-$ in $\bar{\Phi}$ is considered similarly.

We now establish the claim for the three remaining variables, $P^t_{x,C,S}$, $P^t_{x,C,U}$, $P^t_{x,C,*}$. Note that the sets of clauses $\mathcal{S}_{x,C}$ and $\mathcal{U}_{x,C}$ are the same for the formulas Φ and $\bar{\Phi}$. Applying (26) to compute $P^t_{x,C,U}$, using the inductive assumption $P^{t-1}_{C,x,S} = Q^{t-1}_{C,x,U}$, $P^{t-1}_{C,x,U} = Q^{t-1}_{C,x,S}$, and comparing with (27), we see that $P^t_{x,C,S} = Q^t_{x,C,U}$. Similarly, we see that $P^t_{x,C,U} = Q^t_{x,C,S}$. Finally, applying (28), we see that $P^t_{x,C,*} = Q^t_{x,C,*}$. This completes the proof of the induction.

Now define $Z_x(0)$, $Z_x(1)$, and $Z_x(*)$ in terms of P^r in the same way as $W_x(0)$, $W_x(1)$, and $W_x(*)$ are defined in terms of Q^r, namely via identities (29), (30), and (31). Again we see that $Z_x(0) = W_x(1)$, $Z_x(1) = W_x(0)$, and $Z_x(*) = W_x(*)$, further implying $\mathbb{P}(Z_x(1) > Z_x(0)) = 1 - \mathbb{P}(W_x(1) > W_x(0))$. Thus the rule $\tau(B_\Phi(x,r)) = \mathbb{P}(W_x(1) > W_x(0))$ is balanced. ∎

Acknowledgments. The authors gratefully acknowledge many enlightening conversations with Federico Ricci-Tersenghi, Riccardo Zecchina, Marc Mezard, Giorgio Parisi, Florent Krzakala, Lenka Zdeborova, and many others who generously provided constructive feedback on the earlier version of this paper.

REFERENCES

[ACO08] D. Achlioptas and A. Coja-Oghlan, *Algorithmic barriers from phase transitions*, in Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS'08), IEEE, Piscataway, NJ, 2008, pp. 793–802.

[ACORT11] D. Achlioptas, A. Coja-Oghlan, and F. Ricci-Tersenghi, *On the solution space geometry of random formulas*, Random Structures Algorithms, 38 (2011), pp. 251–268.

[AKKT02] D. Achlioptas, J. H. Kim, M. Krivelevich, and P. Tetali, *Two-coloring random hypergraphs*, Random Structures Algorithms, 20 (2002), pp. 249–259.

[AM06] D. Achlioptas and C. Moore, *Random k-SAT: Two moments suffice to cross a sharp threshold*, SIAM J. Comput., 36 (2006), pp. 740–762, https://doi.org/10.1137/S0097539703434231.

[BMZ05] A. Braunstein, M. Mézard, and R. Zecchina, *Survey propagation: An algorithm for satisfiability*, Random Structures Algorithms, 27 (2005), pp. 201–226.

[CO10] A. Coja-Oghlan, *A better algorithm for random k-SAT*, SIAM J. Comput., 39 (2010), pp. 2823–2864, https://doi.org/10.1137/09076516X.

[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, Philadelphia, 2011, pp. 957–966, https://doi.org/10.1137/1.9781611973082.74.

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

[DRZ08] L. DallAsta, A. Ramezanpour, and R. Zecchina, *Entropy landscape and non-Gibbs solutions in constraint satisfaction problems*, Phys. Rev. E, 77 (2008), 031118.

[GS14] D. Gamarnik and M. Sudan, *Limits of local algorithms over sparse random graphs*, Ann. Probab., to appear.

[Het16] S. Hetterich, *Analysing Survey Propagation Guided Decimation on Random Formulas*, preprint, https://arxiv.org/abs/1602.08519, 2016.

[HLS] H. Hatami, L. Lovász, and B. Szegedy, *Limits of locally-globally convergent graph sequences*, Geomet. Funct. Anal., 24 (2014), pp. 269–296.

[Kar76] R. M Karp, *The probabilistic analysis of some combinatorial search algorithms*, Algorithms and Complexity, 1 (1976), pp. 1–19.

[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*, Proc. Natl. Acad. Sci. USA, 104 (2007), pp. 10318–10323.

[KSS12] L. Kroc, A. Sabharwal, and B. Selman, *Survey Propagation Revisited*, preprint, https://arxiv.org/abs/1206.5273, 2012.

[Lev86] L. A Levin, *Average case complete problems*, SIAM J. Comput., 15 (1986), pp. 285–286, https://doi.org/10.1137/0215020.

[MM09] M. Mezard and A. Montanari, *Information, Physics and Computation*, Oxford Graduate Texts, Oxford University Press, New York, 2009.

[MMW07] E. Maneva, E. Mossel, and M. J. Wainwright, *A new look at survey propagation and its generalizations*, J. ACM, 54 (2007), 17.

[MPZ02] M. Mézard, G. Parisi, and R. Zecchina, *Analytic and algorithmic solution of random satisfiability problems*, Science, 297 (2002), pp. 812–815.

[MRT11] A. Montanari, R. Restrepo, and P. Tetali, *Reconstruction and clustering in random constraint satisfaction problems*, SIAM J. Discrete Math., 25 (2011), pp. 771–808, https://doi.org/10.1137/090755862.

[NO08] H. N. Nguyen and K. Onak, *Constant-time approximation algorithms via local improvements*, in Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS'08), IEEE Press, Piscataway, NJ, 2008, pp. 327–336.

[RTS09] F. Ricci-Tersenghi and G. Semerjian, *On the cavity method for decimated random constraint satisfaction problems and the analysis of belief propagation guided decimation algorithms*, J. Statist. Mech., 2009 (2009), P09001.

[RV14] M. Rahman and B. Virag, *Local algorithms for independent sets are half-optimal*, Ann. Probab., to appear.