New Bounds for Random Constraint Satisfaction Problems via Spatial Coupling

Dimitris Achlioptas[1] and S. Hamed Hassani, Nicolas Macris, Rudiger Urbanke[2]

[1] UC Santa Cruz, Santa Cruz, CA 95064, USA, optas@cs.ucsc.edu
[2] EPFL-IC-LTHC {seyedhamed.hassani, nicolas.macris, rudiger.urbanke}@epfl.ch

Abstract. This paper is about a novel technique called spatial coupling and its application in the analysis of random constraint satisfaction problems (CSP). Spatial Coupling was recently invented in the area of error correcting codes where it has resulted in efficient capacity-achieving codes for a wide range of channels. However, this technique is not limited to problems in communications. It can be applied in the much broader context of graphical models. We describe here a general methodology for applying spatial coupling to constraint satisfaction problems. We begin by describing how spatially coupled CSPs are constructed. We then use the results of a previous work to argue that spatially coupled CSPs are much easier to solve than standard CSPs while the satisfiability threshold of coupled and standard CSPs are the same. As a result, these features provide a new avenue for obtaining better, provable, algorithmic lower bounds on satisfiability thresholds of the standard (uncoupled) CSP models. We then consider simple algorithms for solving coupled CSPs and provide the necessary machinery to analyze such algorithms. As a consequence, we derive new lower bounds for the satisfiability threshold of standard random CSPs. As we will see, some of these lower bounds surpass the current best lower bounds in the literature.

Key words: algorithms, random structures, constraint satisfaction problems, spatial coupling, threshold saturation, phase transitions

1 Introduction

Constraint satisfaction problems (CSPs) have attracted considerable attention among the communities of computer science, mathematics and statistical physics. Most interesting CSPs, such as satisfiability or coloring, are NP complete, i.e., worst-case inputs are hard.

An alternative question is to consider ensembles of CSPs and to investigate properties of *typical* instances. For example, one can ask whether a random graph chosen from the Erdos-Renyi ensemble, $G(N, p = \frac{c}{N})$, is Q-colorable or not with high probability. This problem is known as the random Q-COL problem. Another example, is to ask whether a satisfiability formula chosen uniformly at random among all possible formulas consisting of N boolean variables and $m = \lfloor N\alpha \rfloor$ clauses of degree K is satisfiable or not with high probability. This problem is known as the random K-SAT problem.

Consider a random CSP formula, e.g., a random formula from the K-SAT ensemble or a random Erdos-Renyi graph, in the regime where $N \to \infty$. Numerical experiments, physical arguments, as well as the experience from simpler CSPs, indicate that, as the clause density (the value α in satisfiability and c in coloring) crosses a critical threshold, these formulas undergo a phase transition. Below a critical threshold a randomly chosen such formula is almost certainly satisfiable, but above it is almost certainly unsatisfiable. Despite strong evidence, it is yet unproven if such a critical density indeed exists for $K \geq 3$. A considerable effort has gone into providing bounds on such a critical density for different CSPs (we will have more to say about this later on).

We illustrate the use of a novel technique, called *spatial coupling*, for the analysis of constraint satisfaction problems (CSPs). We will show how to use spatial coupling to provide better bounds on the critical threshold. The basic method has three steps. In the first step, given an ensemble of CSPs, we construct from it a new spatially coupled ensemble. In a second step we then derive bounds on the critical threshold for this spatially coupled ensemble. In the final step we show that this bound is also a valid bound on the threshold of the underlying (not coupled) ensemble.

Our method is quite general and should apply to a wide variety of CSPs. But for the purpose of exposition we apply our method to a concrete example, namely to the K-SAT problem. We derive new lower bounds for the satisfiability threshold of random CSPs. As we will see, some of these lower bounds surpass the current best lower bounds in the literature. Indeed, the bounds that we derive in this paper are based on analyzing simple algorithms on the coupled CSP ensembles. Looking ahead, one might hope that more sophisticated (but still analyzable) algorithms for coupled ensembles can provide us with better bounds, perhaps even all the way to the true satisfiability threshold. This is the topic of our current research.

Spatial coupling was initially introduced in the context of channel coding (information theory) where it has resulted in efficient and universal capacity-achieving codes for a wide range of channels which are of interest in communications [1–3]. But spatial coupling is not limited to problems in communications. It can be applied in the much broader context of graphical models. E.g., in signal processing, it has given rise to measurement matrices which, together with low-complexity iterative processing, can achieve the information theoretic bounds for compressive sensing [4–6]. Let us briefly explain the main idea of spatial coupling in this context.

Given a graphical model that represents a "hard" problem (e.g., decoding, inference), we construct from it a larger instance of the same problem but with a particular graphical structure. This structure can be thought of as adding a spatial dimension to the graphical model at hand. Due to this additional structure, the new instance is significantly easier to solve. Mathematically speaking, this change in behavior manifests itself by a significant increase in the threshold under low-complexity processing. In inference problems this "threshold" is a measure of how much "noise" the system can tolerate and still be expected to work correctly. Under optimal processing (which is typically of exponential complexity) a system can tolerate significantly more noise than under low-complexity (message-passing) processing. The curious and important characteristic of spatial coupling is that spatially coupled systems have a threshold under low-complexity processing, which may be as large as the threshold of the underlying system under optimal processing. This phenomenon is named threshold saturation in [1, 2]. The word "saturation" indicates that the threshold under low-complexity processing has increased to its largest possible value, namely the value of optimal processing. In other words, it has saturated. This threshold saturation effect has an obvious and important operational consequence: for a properly designed graphical model, low-complexity processing suffices to reach the optimal performance.

Let us now be more specific and explain the technique of spatial coupling for random CSP ensembles. We begin by some preliminary definitions and notations.

1.1 Preliminaries

The *satisfiability conjecture* states the following: For $K \geq 2$, there exists a constant $\alpha_s(K)$ such that the following holds

$$\lim_{N\to\infty} \Pr\{\mathrm{SAT}(N, K, \alpha) \text{ is satisfiable}\} = \begin{cases} 1 \text{ if } \alpha < \alpha_s(K), \\ 0 \text{ if } \alpha > \alpha_s(K), \end{cases} \tag{1}$$

where by $\mathrm{SAT}(N, K, \alpha)$ we mean the random K-SAT ensemble. The value $\alpha_s(K)$ is called the *satisfiability threshold* of the K-SAT ensemble.

For the Q-COL ensemble, a similar conjecture exists. In more detail, for $Q \geq 2$ it is conjectured that there exists a critical value $c_s(Q)$, called the *colorability threshold*, such that

$$\lim_{N\to\infty} \Pr\{G(N, \frac{c}{N}) \text{ is } Q\text{-colorable}\} = \begin{cases} 1 \text{ if } c < c_s(Q), \\ 0 \text{ if } c > c_s(Q). \end{cases} \tag{2}$$

For the K-XORSAT ensemble, a relation of the type (1) is indeed proven [13]. Let us denote the satisfiability threshold of the K-XORSAT ensemble by $\beta_s(K)$.

For $K, Q = 2$, the satisfiability conjecture is known to be true and we have $\alpha_s(2) = 1$ [7] and $c_s(2) = 1$. The closest analytical result to the satisfiability threshold is a statement due to Friedgut [8]: For $K \geq 3$, there exists a sequence $\{\alpha_s(K,N)\}_{N\in\mathbb{N}}$ such that for any $\epsilon > 0$ the following holds

$$\lim_{N\to\infty} \Pr\big\{\mathrm{SAT}(N,K,\alpha) \text{ is satisfiable}\big\} = \begin{cases} 1 & \text{if } \alpha < (1-\epsilon)\alpha_s(K,N), \\ 0 & \text{if } \alpha > (1+\epsilon)\alpha_s(K,N). \end{cases} \tag{3}$$

This statement comes very close to proving the satisfiability conjecture except that the sequence $\alpha_s(K,N)$ is not known to converge to a well-defined limit. In particular, there remains the possibility that such a sequence oscillates in a small window and thus might not converge. A similar result for the coloring problem also holds [9].

Despite the fact that the existence of the satisfiability threshold is yet unproven, there is another "type" of satisfiability threshold whose existence has been proven rigorously [10, 11]. It is known as the max-satisfiability threshold and can be defined informally as follows: Instead of looking at satisfying assignments for a K-SAT formula, let us be content with assignments that violate at most a sub-linear number of clauses.

To this end, the max-satisfiability threshold is a density above which w.h.p no formula has such a "solution." More precisely, consider a random formula F. For an assignment $\underline{x}$ of the variables in F, we define the energy of the assignment, denoted by $H_F(\underline{x})$, to be the number of clauses in F that the assignment $\underline{x}$ violates. For the formula F, we define its minimum energy level or ground state, $\mathcal{H}_F$, to be the minimum possible energy that can be reached for F over all the assignments $\underline{x}$, i.e.,

$$\mathcal{H}_F = \min_{\underline{x}\in\{0,1\}^N} H_F(\underline{x}). \tag{4}$$

It is more convenient to work with the normalized version of the ground state, i.e., $\frac{1}{N}\mathcal{H}_F$. In fact, it can be shown that almost surely (a.s.) for a random formula F, the ground state per variable, $\frac{1}{N}\mathcal{H}_F$, concentrates around its average, and this average converges to a limiting value [10, 11]. That is,

$$\frac{\mathcal{H}_F}{N} \xrightarrow{\text{a.s.}} \lim_{N\to\infty} \frac{\mathbb{E}[\mathcal{H}_F]}{N} \triangleq \mathcal{H}(\alpha, K). \tag{5}$$

For $K \geq 2$ the max-satisfiability threshold, $\alpha_s(K)$, is such that the following holds

$$\mathcal{H}(\alpha, K) \begin{cases} = 0 \text{ if } \alpha < \alpha_s(K), \\ > 0 \text{ if } \alpha > \alpha_s(K). \end{cases} \tag{6}$$

In other words, for $\alpha < \alpha_s(K)$ there is an assignment that satisfies all the clauses except a sub-linear fraction of clauses, whereas for $\alpha > \alpha_s(K)$, any choice of the variable assignments violates a constant fraction of the clauses. Here we have intentionally abused our notation and denoted both the satisfiability threshold and max-satisfiability threshold by $\alpha_s(K)$. We do this, since it is widely believed that these two thresholds coincide.

For Q-COL there is a similar concept called the max-colorability threshold, which we denote by $c_s(Q)$. Here, we are seeking for coloring assignments (solutions) that violate at most a sub-linear number of edges (constraints). Given a graph G and a coloring assignment $\underline{y}$ of its vertices, we denote by $H_G(\underline{y})$ the number of edges in G such that their two end-points have the same color. The minimum energy of G is then defined as

$$\mathcal{H}_G = \min_{\underline{y}\in\{0,1,\cdots,Q-1\}^N} H_G(\underline{y}). \tag{7}$$

It can be shown that if G is a random graph from $G(N, c)$, then the quantity $\frac{\mathcal{H}_G}{N}$ concentrates around its average, and this average has a well defined limit denoted by $\mathcal{H}(c, Q)$. The max-colorability threshold, $c_S(Q)$, is then defined as the value where this normalized minimum energy is non-zero w.h.p:

$$\mathcal{H}(c, Q) \begin{cases} = 0 \text{ if } c < c_s(Q), \\ > 0 \text{ if } c > c_s(Q). \end{cases} \tag{8}$$

Again it is widely believed that the colorability and max-colorability thresholds coincide for any value of Q and hence we use the same notation for both. For the K-XORSAT problem it is proven [13] that these two types of threshold coincide.

This is *important:* Throughout this paper, we will only consider the max-satisfiability threshold and whenever we talk about $\alpha_s(K)$, or $c_s(Q)$, we mean the max-satisfiability or the max-colorability thresholds defined in (6) and (8), respectively. We also denote the (max-)satisfiability threshold of the K-XORSAT ensemble by $\beta_s(K)$.

1.2 Coupled CSPs

Recall that our program has three steps, the first of which is to introduce *coupled* ensembles.

Coupled K-SAT Ensembles: This ensemble represents a chain of coupled underlying K-SAT ensembles. Figure 1 is a visual aid but gives only a partial view. We consider L clause positions $z \in \{0, 1, \cdots, L-1\}$ and $L + w - 1$ variable positions $z \in \{0, 1, \cdots, L + w - 2\}$. At each variable position z, we lay down N Boolean variables. Also, for each check position z, we lay down $M = \lfloor \alpha N \rfloor$ clauses of length K. So in total we have NL variables and ML clauses. Each clause c at a position z, chooses each of its K variables via the following procedure. We first choose a position $z + j$ with j picked uniformly at random from the window $\{0, \cdots, w - 1\}$, then we pick a variable uniformly at random among all the N variables located at position $z + j$, and finally we connect the clause c and the variable. All the K variables of the clause c are chosen independently in this way. The sign of each edge is chosen independently by flipping a fair coin. This ensemble is called the (spatially) coupled K-SAT ensemble and an instance of it is called a coupled formula. We denote such an ensemble by CSAT(N, K, α, w, L).

It is natural to extend the concepts developed for the K-SAT ensemble to the coupled K-SAT ensemble. As the coupled ensemble has two additional parameters, namely L and w, our notation for the coupled ensemble will include an additional L and w in the subscript. We denote the satisfiability threshold of the coupled ensemble by $\alpha_{s,L,w}(K)$ and also denote the ground state per variable of the coupled ensemble (as in (5)) by $\mathcal{H}_{L,w}(\alpha, K)$. The overall clause density of this ensemble is $\alpha(1 - \frac{w-1}{L})$, which tends to α as L grows large (and $L \gg w$). It is easy to see that at the positions in the (left and right) boundaries of the chain the variables have a smaller average degree compared to the positions in the middle. Hence, the formula has been terminated at the boundaries in such a way that it is easier to solve at positions close to the boundaries. Moreover, the two ensembles have locally the same structure.

Our main objective is to show that due to the additional spatial structure of the coupled ensemble and the special termination at the boundaries, a new set of remarkable aspects emerge.

Coupled Erdos-Renyi Random Graphs: Coupled Q-COL formulas are constructed in the same manner as the coupled SAT formulas. Again, we have additional parameters L and w. We consider $L + w - 1$ variables positions $z \in \{0, 1, \cdots, L + w - 2\}$, and we place N variables in each position which we label them by a pair (z, i), where $i \in \{0, 1, \cdots, N - 1\}$. Hence, in total we have NL variables. We let $\big((z, i), (z', i')\big)$ denote an edge that connects the variable (z, i) to the variable (z', i'). Given the window size w, we call an edge $\big((z, i), (z', i')\big)$ *possible* if $|z - z'| \leq w$. That is, we only consider edges that connect

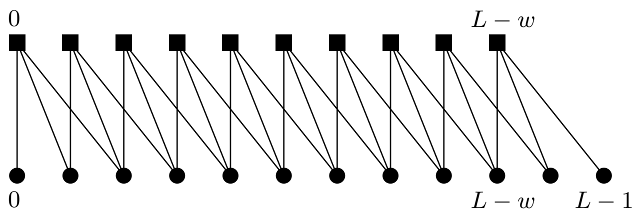

Fig. 1. A representation of the geometry of the graphs with window size $w = 3$ along the "longitudinal chain direction" z. The "transverse direction" is viewed from the top. At each position there is a stack of N variable nodes (circles) and a stack M constraint nodes (squares). The depicted links between constraint and variable nodes represent stacks of edges.

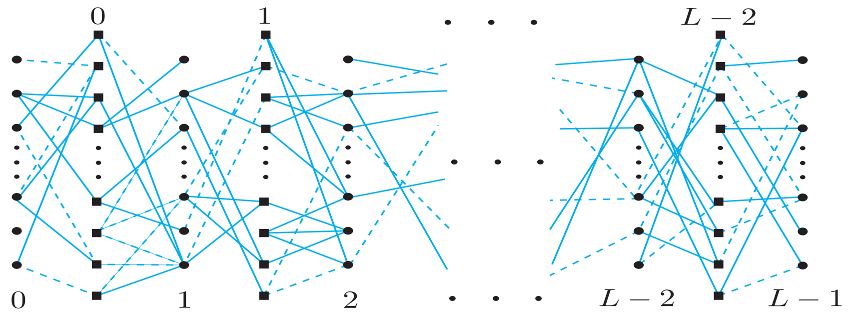

Fig. 2. A coupled 3-SAT formula with $w = 2$. Each clause at a generic position z, is connected (uniformly at random) to variables at positions z or $z + 1$. Here, a dashed edge represents negation of the corresponding variable.

variables whose position differs at most a value w. In order to construct a random coupled graph, we take each of the possible edges and include it in the graph with probability $p = \frac{c}{NL}$ and independently of all the other edges. Figure 3 is an element of such an ensemble when $w = 3$. We denote this ensemble by $G(N, L, w, p = \frac{c}{N(L+w-1)})$. Given an integer $Q \geq 3$, we let $c_{s,L,w}(Q)$ denote the Q-colorability threshold of $G(N, L, w, p = \frac{c}{N(L+w-1)})$.

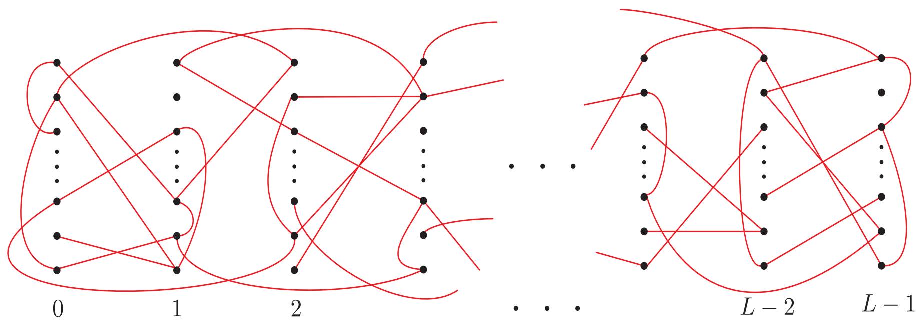

Fig. 3. A coupled Erdos-Renyi graph with chain length L and window length $w = 3$.

Coupled K-XORSAT Ensembles: This ensemble is constructed in the same way as the coupled K-SAT ensemble. We denote by $\beta_{s,L,w}(K)$ the satisfiability threshold of the coupled K-XORSAT ensemble with parameters L, w.

Remark 1. Following along the same lines as above, one can consider a coupled version of other CSPs such as NAE-SAT or $(2+p)$-SAT. For the sake of brevity, we do not go further into details for these CSPs and only repeat the main theme: Align L sets of variables and constraints in a chain, and connect the constraints of each of the sets with the variables in the sets with position not further than w in the chain. There are less constraints at the boundary so that the problem is (locally) easier there.

A few important relations between the coupled and uncoupled ensembles have already been stated in [14]. We will now briefly review these facts and explain their relevance for the current work:

1. The coupled and uncoupled ensembles have the same satisfiability threshold (as $L \to \infty$). This is stated more formally in Theorem 1 that appears shortly in the following. Consequently, any lower bound on $\alpha_{s,L,w}$ can be turned into a lower bound for α_s by taking $L \to \infty$. In particular, any algorithmic lower bound on $\alpha_{s,L,w}$ is guaranteed to be lower bounds for α_s.
2. Statistical physics computations indicate that coupled formulas are significantly easier to solve in a broad range of clause densities. More precisely, for the coupled ensemble the values of α for which the space of solutions is fragmented into well separated clusters are substantially larger compared to the values for uncoupled ensembles. Therefore, we can hope that a form of algorithmic threshold saturation, or at least algorithmic threshold increase, happens when well chosen algorithms are applied to coupled CSPs. This results in proving better algorithmic lower bounds on $\alpha_{s,L,w}$ and thus on α_s. The proposed methodology is our main motivation for this paper.

Theorem 1. *([14]) Let $\alpha_{s,L,w}(K)$ be the max-satisfiability threshold of the coupled K-SAT ensemble with chain length L and coupling length w. Also let $\alpha_s(K)$ denote the max-satisfiability threshold of the standard (un-coupled) K-SAT ensemble. We have*

$$\alpha_s(K) = \lim_{L\to+\infty} \alpha_{s,L,w}(K). \tag{9}$$

A similar result holds for the Q-COL ensembles. Denoting max-colorability threshold of the coupled Q-COL ensemble by $c_{s,L,w}(Q)$ and also the threshold of the standard ensemble by $c_s(Q)$, we have

$$c_s(Q) = \lim_{L\to+\infty} c_{s,L,w}(Q). \tag{10}$$

For the K-XORSAT ensemble (with K even), we also have

$$\beta_s(K) = \lim_{L\to+\infty} \beta_{s,L,w}(K). \tag{11}$$

1.3 Statement of the Main Results

The main objective of this paper is to investigate the algorithmic consequences of spatial coupling. In detail, we consider two classes of algorithms to solve CSP formulas, namely peeling-type algorithms and decimation algorithms, and analyze their behavior on the coupled ensembles. As we will see, for these algorithms the effect of spatial coupling can be described by a relevant threshold saturation phenomenon. Also, as mentioned above, the satisfiability thresholds of the coupled and standard (uncoupled) CSP ensembles are the same. Hence, we obtain a lower bound on the satisfiability threshold of the *standard* CSP ensemble by computing the highest constraint density for which an specific algorithm is successful in finding a satisfying assignment of the *coupled* CSP ensemble.

In order to better illustrate the idea of spatial coupling and its algorithmic consequences, let us consider a concrete example: The K-XORSAT ensemble that is the simplest instance among the class of constraint satisfaction problems. In order to find a solution for the XORSAT problem, the belief propagation algorithm is equivalent to a simple peeling (leaf-removal) procedure. For the standard K-XORSAT ensemble, it is known that this peeling algorithm succeeds with high probability in finding an assignment as long as the clause density of the formula is less than a critical value $\beta_d(K)$. It is also widely believed that $\beta_d(K)$ is the point where standard message passing algorithms fail to find solutions (e.g., the belief propagation algorithm). Therefore, the regime of the clause densities inside the interval $(\beta_d(K), \beta_s(K))$ can be considered as the hard-SAT regime.

Now, consider the coupled K-XORSAT ensemble. As a consequence of the coupling there is a remarkable performance improvement under message-passing. This improvement is due to the special termination, as well as to the special spatial structure. For instance, consider the peeling algorithm. Due to the extra help at the boundary, the problem can be partially solved there, even when using a (suboptimal) peeling algorithm. This in turn makes the problem easier for the neighboring copies. This effect cascades over the whole length of the chain. A simple, albeit not very accurate, analogy is a chain of properly placed domino pieces. Once we topple a boundary piece, the whole chain is toppled. But contrary to domino pieces, where forces act in only one direction, in spatially coupled graphs there is true interaction, and information bounces back and forth between neighboring systems. It turns out that the peeling algorithm performs well on the coupled formulas all the way up to the satisfiability threshold $\beta_s(K)$. Also, from Theorem 1 we know that both the coupled and uncoupled ensembles have the same satisfiability threshold. Thus, by only analyzing the peeling algorithm over the coupled ensemble we can obtain a matching lower bound on the satisfiability threshold of the coupled as well as the uncoupled (standard) ensembles. Perhaps the most surprising aspect of spatial coupling is not that the performance is simply improved, but that such systems perform under low-complexity message passing algorithms as well as if they had been solved optimally, i.e., their performance saturates.

We proceed by explaining in detail the main results of the paper. For brevity, we illustrate the results in detail for the random K-SAT problem, and mention briefly similar results for the random Q-COL and K-XORSAT problems. For a K-SAT formula, the pure literal algorithm is perhaps the simplest known algorithm for solving satisfiability problems. It works up to a critical density $\alpha_{\text{pl}}(K)$ where a non-trivial core emerges inside the formula. For an uncoupled formula, this critical density can be found with the help of a simple scalar fixed point equation. This algorithm extends naturally to coupled formulas. Its critical density $\alpha_{\text{pl},L,w}$, where the non-trivial core develops, is found by analyzing the fixed points of a set of one-dimensional fixed point equations. The recent one-dimensional theory of [17] and [18] enables us to analytically compute the limiting value of $\alpha_{\text{pl},L,w}$ when $L \gg w \gg 1$. Let us denote this limit by $\alpha_{\text{cpl}}(K)$; i.e.,

$$\alpha_{\text{cpl}}(K) = \lim_{w\to\infty} \lim_{L\to\infty} \alpha_{\text{pl},L,w}(K).$$

The last two rows of Table 1 include the corresponding thresholds for coupled and uncoupled ensembles. For large K, we find [3]

$$\alpha_{\text{pl}}(K) \doteq \frac{2\ln K}{K} \text{ but } \alpha_{\text{cpl}}(K) \doteq 2.$$

[3] For two sequences $\{a_n\}$ and $\{b_n\}$, we say $a_n \doteq b_n$ if $\frac{a_n}{b_n} \to 1$ as $n \to \infty$.

Hence, there is roughly a factor $\frac{K}{\ln K}$ of threshold improvement via spatial coupling. However, the coupled threshold of this algorithm is still far below the satisfiability threshold (which is a constant away from $2^K \ln 2$).

K	3	4	5	large K
bestlowerbound	3.52	7.91	18.79	$2^K \ln 2 - \Theta(1)$
$\alpha_{\text{cuc}}(K)$	3.67	7.81	15.76	2^{K-1}
$\alpha_{\text{uc}}(K)$	2.66	4.50	7.58	$\frac{e2^{K-1}}{K}$
$\alpha_{\text{cpl}}(K)$	1.834	1.954	1.986	2
$\alpha_{\text{pl}}(K)$	1.626	1.544	1.402	$\frac{2\ln K}{K}$

Table 1. Thresholds for the peeling and unit clause propagation algorithm corresponding to the coupled and un-coupled K-SAT ensembles. The first line corresponds to best lower bounds for K-SAT in the literature: The best previous lower bound for $K=3$ is due to [31, 32]. For $K>3$ the best lower bounds are due to [29]. The asymptotic bound for large K is due to [28–30].

We next consider the unit clause propagation (UC) algorithm that is the simplest type of decimation algorithm. We first derive a suitable schedule to perform the decimation steps and then we develop the required machinery to analyze this decimation algorithm on the coupled formulas (see also Theorem 4). Let us denote by $\alpha_{\text{uc}}(K)$ and $\alpha_{\text{uc},L,w}(K)$ the thresholds of the UC algorithm for the individual and coupled ensembles, respectively. We also define

$$\alpha_{\text{cuc}}(K) = \lim_{w\to\infty} \lim_{L\to\infty} \alpha_{\text{uc},L,w}(K).$$

Table 1 contains the corresponding thresholds for the coupled and individual ensembles. For large K we find that

$$\alpha_{\text{uc}}(K) \doteq \frac{e2^{K-1}}{K} \;\text{ but }\; \alpha_{\text{cuc}}(K) \doteq 2^{K-1}.$$

Again, the coupled threshold is improved roughly by a factor $\frac{K}{e}$ over the individual threshold. There are a few interesting comments in order:

For small K the coupled thresholds of the UC algorithm are comparable to the best lower bounds in the literature. For $K=3$, the value 3.67 is a new lower bound for the K-SAT problem[4]. However, as K grows, these coupled thresholds become smaller than the best proven lower bounds.

As mentioned above, the threshold of the coupled UC algorithm is asymptotically 2^{K-1}. This intuitively confirms the fact that for the coupled ensembles, the dynamical transition is well above the dynamical transition of the individual K-SAT ensemble. In other words, the coupled formulas can be considered "easier" than the uncoupled ones in the sense that they break well through the algorithmic barrier $\frac{2^K \ln K}{K}$ of the individual ensemble. We believe that more sophisticated (and analyzable) algorithms for the coupled formulas can provide us with better bounds and it is even plausible to talk about algorithms that succeed all the way up to the satisfiability threshold. This is the topic of our current research.

For the random Q-COL problem, similar results as above hold analytically (see Table 2). The peeling algorithm works as long as the corresponding graph has an empty Q-core. For large Q we have

$$c_{\text{pl}}(Q) \doteq Q \;\text{ but }\; c_{\text{cpl}}(Q) \doteq 2Q. \tag{12}$$

[4] To be more precise, this is a new lower bound for the 3-MAX-SAT problem.

Also, an algorithm similar to the UC algorithm is proposed for the coupled Q-COL model. For $Q = 3$, this algorithm finds a proper coloring for average connectivity values up to $c_{\text{cuc}}(3) = 4.44$. We note from [12] that the condensation threshold for $Q = 3$ is $c_c(3) = 4$, which is below $c_{\text{cuc}}(3)$. Hence, for the coupled ensemble we are capable to go even above the condensation threshold. However, we are sill below the freezing threshold. The SAT/UNSAT threshold for $Q = 3$ is $c_s(3) = 4.69$. For large Q, we find that $c_{\text{cuc}}(Q) \doteq 2Q\log(Q) - Q$.

Q	3	4	large Q
best lower bound	4.03	8.3	$2Q\ln Q - \ln Q - \Theta(1)$
$c_{\text{cul}}(Q)$	4.44	8.23	$2Q\ln Q - Q$
$c_{\text{cpl}}(Q)$	3.58	5.74	$2Q$
$c_{\text{pl}}(Q)$	3.35	5.14	Q

Table 2. Thresholds for the peeling and unit clause propagation algorithm corresponding to the coupled and un-coupled Q-COL ensembles. The first line corresponds to best lower bounds for Q-COL in the literature: The best previous lower bound for $Q = 3$ is due to [34]. For $Q > 3$ the best lower bounds are due to [35]. The asymptotic bound for large Q is due to [35, 36].

The peeling (leaf removal) thresholds for K-XORSAT individual and coupled ensembles, $\alpha_{\text{lr}}(K)$, $\alpha_{\text{lr},L,w}(K)$, are obtained just by halving the K-SAT pure literal thresholds. Interestingly enough, it can be shown that the coupled threshold of the leaf removal algorithm is precisely equal to the SAT/UNSAT threshold for the K-XORSAT problem. That is, $\lim_{w\to\infty}\lim_{L\to\infty}\beta_{\text{lr},L,w}(K) = \beta_s(K)$. Hence, we have a full threshold saturation for the K-XORSAT problem.

The rest of this paper is organized as follows: The peeling algorithms together with their analysis are described in Section 2. In Section 3 we describe decimation algorithms and the required machinery for their analysis on the coupled CSPs.

2 Peeling Algorithms

Below we illustrate a simple peeling-type algorithm applied to K-SAT, which in the literature is called the pure literal algorithm. We then discuss briefly similar algorithms for Q-COL and K-XORSAT. As we will see in the sequel, such peeling algorithms can be cast into the framework of one-dimensional coupled recursions for which a recent elegant characterization has been provided in [17] and [18].

2.1 Pure Literal: A Peeling Algorithm for K-SAT

We begin by a brief explanation of the algorithm. Let G be a K-SAT formula. The algorithm starts with G and in each step shortens G until we either reach the empty graph or we cannot make any further shortening. Assume now that there exists a literal (variable) i in G such that all of its incoming edges have the same sign. This literal is called a *pure literal*. One can choose a value for x_i that satisfies all of its neighboring clauses and clearly that is the optimum (and obvious) choice to fix the variable i in order to find a SAT assignment for G. Hence, without loss of generality, we can remove i and its neighboring clauses from G and search for a SAT assignment on the graph $G \setminus i$. In other words, finding a SAT assignment for G is reduced to finding an assignment for $G \setminus i$. As a result, we can peel the literal i and its neighbors from G and look for new pure literals on $G \setminus i$. We continue this process until no further progress is possible. This happens when the residual graph is either empty or if it has no more pure literals. If the final graph is empty then the algorithm succeeds; otherwise, it fails. This algorithm determines has been analyzed by the method of differential equations [37]. Here we discuss the algorithm from the message passing point of view.

Consider the following message passing (MP) rule. As we see later, this MP rule is equivalent to the pure literal algorithm. At time $t \in \{1, 2, \cdots\}$, assign to each edge $\langle i, c\rangle \in E$ two messages $\mu^t_{i\to c}$ and $\mu^t_{c\to i}$. The messages $\mu^t_{i\to c}$ represent the messages going from literals to clauses at time t and the messages $\mu^t_{c\to i}$ are the messages from checks to variables at time t. The messages at time $t+1$ are evolved from the ones at time t via the following procedure:

1. At time 0, initialize all the messages $\mu^0_{c\to i}$ and $\mu^0_{i\to c}$ to 0.
2. At time $t+1$,

$$\mu^{t+1}_{c\to i} = \mathbb{1}\{\sum_{j\in\partial c\setminus i} \mu^t_{j\to c} \geq 1\},$$
$$\mu^{t+1}_{i\to c} = \prod_{h\in\partial i\setminus c, \mu^{t+1}_{h\to i}=0} \mathbb{1}\{J_{c,i} = J_{h,i}\}.$$

Here, we recall that $J_{c,i}$ denotes the sign of the edge $\langle c, i\rangle$. The above message passing rule is equivalent to the pure literal (peeling) algorithm in the following sense. When $\mu^t_{i\to c} = 1$ for at least one $c \in \partial i$, then the vertex i would have been peeled by the algorithm some time before t and if $\mu^t_{i\to c} = 0$, the vertex i would not have been peeled by the algorithm up to time t. The same statement is valid for the clauses in a way that if $\mu^t_{c\to i} = 1$ for at least one $i \in \partial c$, then the clause c would have been peeled at some time before t in the pure literal algorithm.

Define $p^t = \mathbb{P}(\mu^t_{c\to i} = 0)$ and $q^t = \mathbb{P}(\mu^t_{i\to c} = 0)$. Note that $p^0 = 1$. We derive the density evolution equations that relates p^{t+1} to p^t. Let G be randomly chosen from $\text{SAT}(N, K, \alpha)$ with N very large. Fix an edge $\langle c, i\rangle$. Observe that $\mu^{t+1}_{c\to i} = 0$ if and only if all the incoming messages to the clause c, other that the one of $\langle c, i\rangle$, take value 0. Hence, we can write

$$p^{t+1} = (q^t)^{K-1}. \tag{13}$$

Relating q^{t+1} to p^t is slightly more subtle. Observe that $\mu^{t+1}_{i\to c} = 1$ if and only if the sign of $\langle c, i\rangle$ is equal to the sign of every edge $\langle h, i\rangle$ such that $h \in \partial i \setminus c$ and $\mu^{t+1}_{h\to i} = 0$. Moreover, the probability that an edge $\langle c, i\rangle$ is incident to a variable i of degree d is $\frac{e^{-\alpha K}(\alpha K)^{d-1}}{(d-1)!}$. One can then write

$$\begin{aligned}
1 - q^{t+1} &= \sum_{d=1}^{\infty} \frac{e^{-\alpha K}(\alpha K)^{d-1}}{(d-1)!} \sum_{j=0}^{d-1} \binom{d-1}{j} (p^{t+1})^j (1-p^{t+1})^{d-1-j} 2^{-j} \\
&= \sum_{d=1}^{\infty} \frac{e^{-\alpha K}(\alpha K)^{d-1}}{(d-1)!} (1 - \frac{p^{t+1}}{2})^{d-1} \\
&= \exp(-\frac{\alpha k}{2} p^{t+1}).
\end{aligned}$$

Hence, from the above two relations we obtain that

$$p^{t+1} = (1 - \exp(-\frac{\alpha K}{2} p^t))^{K-1}, \tag{14}$$

with $p^0 = 1$. It is more convenient to do a change of variables in the form of $x^t = \alpha p^t$, which transforms (14) to

$$x^{t+1} = \alpha(1 - \exp(-\frac{K}{2} x^t))^{K-1}. \tag{15}$$

For the pure literal algorithm to succeed, the value of x^t should tend to 0 and t increases. Now, from the recursion (15) and fact that $x_0 = 1$, it is easy to deduce that for t growing large, x^t tends to 0 if and only if the equation

$$x = \alpha(1 - \exp(-\frac{K}{2}x))^{K-1} . \tag{16}$$

has only one solution which is the trivial solution $x = 0$ on $[0, 1]$. The net result is that the pure literal rule succeeds w.h.p for $\alpha < \alpha_{\rm pl}(K)$ such that (16) has a unique fixed point $x = 0$ in the unit interval $[0, 1]$. Mathematically we can define $\alpha_{\rm pl}$ as

$$\alpha_{\rm pl}(K) = \sup\{\alpha \geq 0 \mid x - \alpha(1 - \exp(-\frac{K}{2}x))^{K-1} > 0 \;\; \forall x \in (0, 1]\}. \tag{17}$$

We now consider the coupled ensemble. The way the pure literal algorithm works on a coupled formula is similar to what we explained above and hence needs no further explanation. In order to analyze the pure literal rule we can think of extending the chain to $\mathbb{Z}$ with "pure" variable nodes for $z < 0$ and $z > L+w-2$. The peeling of constraints attached to pure nodes will propagate inside the chain as long as α is below the critical threshold. A similar message passing analysis as above yields a set of one-dimensional coupled recursions

$$x_z^{t+1} = \alpha\Big\{\frac{1}{w}\sum_{l=0}^{w-1}(1 - \exp(-\frac{K}{2w}\sum_{k=0}^{w-1} x_{z+k-l}^t)\Big\}^{K-1}, \tag{18}$$

with boundary condition $x_z^t = 0$ for z at the boundaries. This recursion results in the one-dimensional fixed point equations

$$x_z = \alpha\Big\{\frac{1}{w}\sum_{l=0}^{w-1}(1 - \exp(-\frac{K}{2w}\sum_{k=0}^{w-1} x_{z+k-l})\Big\}^{K-1}. \tag{19}$$

One can define $\alpha_{{\rm pl},L,w}(K)$ as the largest value of α such that (19) has only one fixed point profile to be the all-zero profile.

Table 3 contains the numerical prediction of $\alpha_{{\rm pl},L,w}(K)$ for $L = 80$ and $w = 5$ and different values of K. As we observe from Table 3 there is an improvement of the coupled threshold over the individual ensemble. For example for $K = 3$ we have $\alpha_{\rm pl} \approx 1.636 < \alpha_{\rm pl,w=5,L=80} \approx 1.835 < \alpha_s \approx 4.26$, a modest improvement. As we will see in the sequel, the coupled thresholds $\alpha_{{\rm pl},L,w}(K)$ when $L, w \to \infty$ can be

K	3	4	5	7
$\alpha_{\rm pl}(K)$	1.626	1.544	1.402	1.190
$\alpha_{{\rm pl},L=80,w=5}(K)$	1.834	1.954	1.986	1.998

Table 3. *First line*: Pure threshold for the uncoupled case. *Second line*: Pure literal threshold for a coupled chain with $w = 5$, $L = 80$. By halving these numbers, we obtain the corresponding thresholds of the leaf removal algorithm devised for the K-XORSAT problem.

precisely and analytically computed. We postpone further arguments on the amount of the improvement of the coupled thresholds to Section 2.3.

2.2 Peeling Algorithms for Q-COL and K-XORSAT

We discuss now a similar peeling algorithm for Q-COL. We start with a graph G to color with a given set of Q colors. Assume there exists a node i in G that has degree less than Q. Clearly, if we can color the graph $G \setminus i$ with Q colors, then G can also be colored with Q colors. Hence, finding a Q-coloring for G is equivalent to finding a Q-coloring for $G \setminus i$. As a result, we can peel the node i from G and continue this process until the final graph has no more nodes of degree less than Q. If the final graph is empty then the algorithm succeeds; and otherwise it fails. Such a peeling algorithm can be analyzed in the same way as above. In particular, let us define $y = cx$, where x is the fraction of nodes in the final residual graph and c is the average vertex degree. We find

$$y = c\,G(y), \tag{20}$$

where the function G is given as

$$G(y) = 1 - e^{-y} \sum_{j=0}^{Q-2} \frac{y^j}{j!}. \tag{21}$$

For $c < c_p$ there is a unique trivial fixed point $y = 0$ and the algorithm succeeds. Non trivial fixed points appear for $c > c_p$ which is the threshold for the emergence of a Q-core. Table 4 contains the numerical values of c_{p} for several values of Q.

We now take coupled instances from the ensemble. We can write down the density evolution equations and the corresponding one-dimensional fixed point equations. Not surprisingly, similar calculations show that the message passing algorithm is controlled by the one-dimensional fixed point equation,

$$y_z = c\,G\Big(\frac{1}{2w-1} \sum_{k=-w+1}^{w-1} y_{z+k}\Big). \tag{22}$$

where $y_z = cx_z$ and x_z is the fraction of remaining nodes at position z. Table 4 contains the numerical values of $c_{\mathrm{p,w=5,L=80}}$ for several values of Q, and shows the threshold improvement.

Q	3	4	5	7
c_s	4.69	8.90	13.69	24.46
c_{p}	3.35	5.14	6.79	9.87
$c_{\mathrm{p},L=80,w=5}$	3.58	5.74	7.84	11.92

Table 4. *First line*: static phase transition threshold for Q-COL. *Second line*: peeling algorithm threshold for the uncoupled case. *Third line*: peeling algorithm threshold for a coupled chain with $w = 5$, $L = 80$.

The peeling algorithm for the K-XORSAT problem is known as the "leaf removal" algorithm. As long as there is a leaf variable node, remove it and remove the attached constraint node with its emanating edges. If this process ends with an empty graph the instance is satisfiable. It is known that this algorithm is equivalent to BP message passing, and the density evolution analysis leads to the fixed point equation

$$x = (1 - \exp(-\alpha K x))^{K-1}. \tag{23}$$

Here, x is interpreted as the probability (when the number of iterations goes to infinity) that a constraint node is not removed. There is a threshold α_{lr} above which (23) has non-trivial fixed points (i.e, the fraction of remaining variables is positive), so we get a lower bound $\alpha_{\mathrm{lr}} < \alpha_s$. For the coupled ensemble the density evolution analysis yields the one-dimensional fixed point equations

$$x_z = \left\{ \frac{1}{w} \sum_{l=0}^{w-1} (1 - \exp(-\frac{\alpha K}{w} \sum_{k=0}^{w-1} x_{z+k-l}) \right\}^{K-1}, \tag{24}$$

Note here that these fixed-point equations are equal to the (16) and (19) with the replacement $\alpha \to 2\alpha$. As a result, by halving the numbers in Table 3 we obtain the corresponding thresholds for the leaf-removal algorithm.

2.3 The Framework of Coupled Scalar Recursions

One-dimensional coupled recursions such as (18) have recently been fully characterized in [17] and [18]. Let us now briefly mention the main results in this regard.

A scalar recursion as in (15) can be written in the general form of

$$x^{t+1} = f(g(x^t;\alpha)), \tag{25}$$

where $f : [0,1] \times \mathbb{R} \to [0,1]$ is strictly increasing in both arguments for $x, \alpha > 0$ and $g : [0,1] \to [0,1]$ satisfies $g'(x) > 0$ for $x \in [0,1]$. Such a recursion with the above mentioned properties for f and g is called an *scalar admissible system*. For example, in the recursion (15) we have $f(x;\alpha) = \alpha(1+x)^{K-1}$ and $g(x) = -\exp(-\frac{K}{2}x)$. Hence (15) is a scalar admissible system. The threshold of such scalar system , α_{sys} , is defined as the largest value of α such that the equation $x = f(g(x;\alpha))$ has the unique $x = 0$ fixed point for $x \in [0,1]$.

Let us now consider the coupled system of scalar recursions. Consider the position set $\mathcal{L} = \{0, 1, \cdots, L+w-1\}$ for which we assign variable $x_z^t \in [0,1]$ to $z \in \mathcal{L}$ for time parameter $t \in \{0, 1, \cdots\}$. Also, let us define $x_z^t = 0$ for $z \notin \mathcal{L}$ and all times $t \geq 0$. For the coupled system we have the recursions

$$x_z^{t+1} = \frac{1}{w}\sum_{l=0}^{w-1} f(\frac{1}{w}\sum_{k=0}^{w-1} g(x_{z+k-l}^t);\alpha), \tag{26}$$

for $z \in \mathcal{L}$. The threshold of the coupled system $\alpha_{\text{sys},L,w}$ is then defined as the largest α for which the coupled system of equations

$$x_z = \frac{1}{w}\sum_{l=0}^{w-1} f(\frac{1}{w}\sum_{k=0}^{w-1} g(x_{z+k-l});\alpha), \tag{27}$$

has a unique trivial fixed point, which is the trivial all-zero fixed point.

The limit of $\alpha_{\text{sys},L,w}$ when $L, w \to \infty$ can be computed from the so called *potential function*, $\phi(x,\alpha)$, that is associated to the scalar admissible system, and is defined as

$$\phi(x,\alpha) \triangleq xg(x) - G(x) - F(g(x),\alpha), \tag{28}$$

where $F(x,\alpha) = \int_0^x f(z;\alpha)\mathrm{d}z$ and $G(x) = \int_0^x g(z)\mathrm{d}z$.

Theorem 2 ([17]). *We have*

$$\lim_{w\to\infty}\lim_{L\to\infty} \alpha_{\text{sys},L,w} = \sup\{\alpha \geq 0 \mid \min_{x\in[0,1]} \phi(x,\alpha) \geq 0\}. \tag{29}$$

Theorem 2 provides a practical way to analytically compute the coupled threshold. It is easy to numerically check that the results of Tables 3 and 4 are very close (up to the fourth decimal) to the numbers obtained from Theorem 2. However, we do not expect to get exactly the same numbers as (29). This is because the thresholds in these tables are found for finite choices of L and w.

From (29), we can find the asymptotic value of the thresholds when K is a very large number. For the K-SAT problem, when $K \to +\infty$ we find

$$\alpha_{\rm pl}(K) \doteq \frac{2\ln K}{K} \;\text{ but }\; \alpha_{{\rm pl},L,w}(K) \to 2 \;\text{ as }\; L \gg w \to +\infty. \tag{30}$$

Thus, the pure literal threshold $\alpha_{\rm pl}(K)$ is "infinitely improved" by coupling. Of course this is far away from the satisfiability threshold $\alpha_s \doteq 2^K \ln 2$.

For the problem of Q-COL we obtain from (29)

$$c_{\rm p}(Q) \doteq Q \;\text{ but }\; c_{{\rm p},L,w}(Q) \doteq 2Q \;\text{ as }\; L \gg w \to +\infty. \tag{31}$$

This has to be compared with $c_s(Q) \doteq 2Q \ln Q$.

The leaf removal thresholds for K-XORSAT individual and coupled ensembles, $\alpha_{\rm lr}(K)$, $\alpha_{{\rm lr},L,w}(K)$, are obtained just by halving the K-SAT pure literal thresholds. Interestingly enough, it can be shown that the coupled threshold of the leaf removal algorithm is precisely equal to the SAT/UNSAT threshold for the K-XORSAT problem. That is, $\lim_{w\to\infty} \lim_{L\to\infty} \alpha_{{\rm lr},L,w}(K) = \alpha_s(K)$. Hence, we have a full threshold saturation for the K-XORSAT problem.

For the problems of K-SAT and Q-COL, although the coupled thresholds improve significantly, they are still far below the static threshold. Hence, we need to consider slightly more sophisticated algorithms. We proceed by entering the realm of decimation algorithms, the simplest of which are the "unit constraint propagation" algorithms. For the K-SAT problem, such type of algorithm is called the *unit clause propagation* algorithm. For the rest of this chapter, our main focus is on the uint clause propagation algorithm for the coupled ensemble. We also mention the final results for the same type of algorithm for Q-COL.

3 Decimation Algorithms

In this section, we consider the the unit clause propagation (UC) algorithm for the satisfiability problem. Similar concepts also hold for the coloring problem, which we omit for the sake of brevity and only mention the final results. Let us begin now with the UC algorithm for solving K-SAT formulas.

3.1 Individual Ensemble

The Unit Clause propagation algorithm, or UC for short, is a (randomized) algorithm which goes through variables one at a time, sets them permanently, and simplifies the formula as it goes along. This algorithm is like a DPLL algorithm but only explores one branch of the search tree. In brief, the algorithm works as follows: Consider a K-SAT formula which we represent by a bipartite graph G consisting of N literals or variable nodes and $M = N\alpha$ clauses or check nodes. The algorithm starts with G and in each step removes several nodes from the graph. The UC algorithm consists of two main steps:

- *Free step:* which is when all the check nodes have degree at least 2. This is the situation where the algorithm is free to do whatever it wants. However, UC does the simplest possible action. It chooses a variable uniformly at random among the currently unset variables and sets it permanently to 0 or 1 again with uniform probability.
- *Forced Step:* which is when we have a unit clause (clause with only one remaining edge). In this situation, we better try to satisfy this clause before it is too late. So in some sense we are forced to fix the variable connected to the unit clause.

Once a variable is set, it is removed from the graph together with the clauses that are satisfied by the variable. Also, there might be some clauses, connected to the variable, that are shortened. This is due to the fact that the assigned value of the variable did not satisfy these clauses. Hence, removing the variable from the graph will cause these yet unsatisfied clauses to have a less degree. A detailed description of the UC algorithm is given in Algorithm 1.

Algorithm 1 Unit Clause Propagation Algorithm

1: Start with a given K-SAT formula G.
2: Repeat until all the variables are set.
3: If G contains unit clauses (forced step), then choose one at random and satisfy it by setting its left variable. Remove or shorten clauses containing this variable.
4: If there are no unit clauses (free step), then choose one variable at random from the unset ones and set it by flipping a coin. Remove or shorten clauses containing this variable.

On the analysis side, the progress of UC can be modeled with differential equations [38]. The net result is that for $\alpha < \alpha_{\text{UC}}(K)$ with α_{UC} given as

$$\alpha_{\text{UC}}(K) = \frac{1}{2}\left(\frac{K-1}{K-2}\right)^{K-2}\frac{2^K}{K} \doteq \frac{e}{2}\frac{2^K}{K}, \tag{32}$$

the UC algorithm finds an assignment that satisfies all but $o(n)$ number of clauses. It can also be shown that for densities below α_{UC}, with positive probability the output assignment satisfies all the clauses.

3.2 Description of UC Algorithm for the Coupled Formulas

Let us now focus on the UC algorithm for the coupled formulas. As for the un-coupled case, the UC algorithm consists of two main steps: free and forced. The operation of the algorithm at a forced step is clear: remove all the unit-clauses until no further unit-clause exists. However, at a free step, depending on how we might want to use the chain structure of the formula, we can have different *schedules* for choosing a free variable. As we will see now, for a coupled formula, the schedule within which we are choosing a variable in a free step is very important[5].

Consider for instance the following naive schedule. At a free step, pick a variable uniformly at random from all the remaining variables and fix it by flipping a coin. Computer experiments indicate that this naive schedule gains no threshold improvement over the un-coupled ensemble. This is not surprising since this schedule does not seem to exploit the spatial (chain) structure of the formula and in some ways it greatly resembles the UC algorithm for the un-coupled ensemble. Hence, in order for the UC algorithm to have a threshold improvement over the coupled ensemble, we need to come up with schedules that exploit to some extent the additional spatial structure of the formula. We proceed by illustrating one such successful schedule.

In the very beginning of the algorithm, all the check nodes have degree K and there are no unit clauses. Hence, we are free to fix the variables in the first few steps of the algorithm. If we fix the variables from the left-most position (i.e., the boundary) we are somehow creating a seed at the boundary of the chain. Continuing this action at the free steps, we will eventually create unit clauses and at these forced steps

[5] For the peeling algorithms mentioned in Section 2, the performance of the algorithm is independent of the schedule of peeling steps.

a natural choice is just to clear all the unit clauses as long as long as they exist. However, when we are confronted with a free step, we will again try to help this seed to grow inside the chain. This can be done again by fixing a variable from the remaining left-most position. Consequently, the schedule that we apply is as follows.

– At a *free step*, pick a variable randomly from the left-most position of the remaining graph and fix it permanently by flipping a fair coin.
– At a *forced step*, we get rid of unit clauses as long as they exist.

Computer experiments show that this schedule indeed exhibits a threshold improvement over the un-coupled ensemble. E.g., for the coupled 3-SAT problem, experiments suggest that the threshold of the UC algorithm is around 3.67. This is a significant improvement compared to the threshold of UC for the un-coupled ensemble which is $\frac{8}{3}$. Of course, one cannot be certain about these numbers until they are confirmed with analytic methods. The right tool to analyze the dynamics of the UC algorithm for the un-coupled ensemble is the method of differential equations (see [38]). The rest of this section focuses on writing the differential equations for the UC algorithm on the coupled ensemble and then analyzing these equations. More specifically, in the next section, we work out these differential equations in detail. Later in the subsequent sections, we simplify these equations and provide a general framework to analyze them. Using this framework, we analytically obtain the threshold of the UC algorithm on the coupled ensembles.

3.3 Analysis of the Evolution of UC via Differential Equations

Phases, Types, and Rounds For the coupled ensemble, the analysis of the evolution of UC is much more involved than the un-coupled ensemble. This is because of the fact that the schedule we have used prefers the left-most variable position in a free step. Hence, the number of variables in different positions will evolve differently. As an example, one can easily see that during the algorithm, the first position that all its variables are set is the left-most position (i.e., position 0). After the evacuation of position 0, position 1 becomes the left-most position of the graph and hence, the second position that becomes empty of variables is position 1. Continuing in this manner, the last position that is evacuated is position $L+w-2$. With these considerations, we consider $L + w - 1$ *phases* for this algorithm (see Figure 4). At phase $p \in \{0, 1, \cdots, L + w - 2\}$, all the variables at positions prior to p have been set permanently and as a result, at a free step we will pick a variable from position p.

This statistical asymmetry in the number of variables at each position also affects the the behavior of the number of check nodes in each position. As a result, we consider *types* for the check nodes. For instance, consider a degree two check node. It is easy to see that the probability that this degree two check node is hit (removed or shortened) is greatly dependent on the position of variables that it is connected to. This means that, dependent on the variable positions to which they are connected, we have different types of degree two check nodes. Clearly, the same statement holds for clauses of degree three, four, etc.

Let us now formally define the ingredients needed for the analysis. The notation we use here is slightly hard to swallow immediately. Thus, for the sake of maximum clarity, we try to uncover the details as smoothly as possible. We consider *rounds* for this algorithm. Each round consists of one free step followed by the forced steps that follow it. More precisely, at the beginning of each round we perform a free step and then we clear out all the unit-clauses as long as they exist (forced steps). We let time t be the number of rounds passed so far. This time variable will be called *round time*. The relation between t and the *natural time* (the total number of permanent fixes) is not linear. We also let $L_i(t)$ be the *number of literals* left in variable position $i \in \{0, 1, \cdots, L - 1\}$.

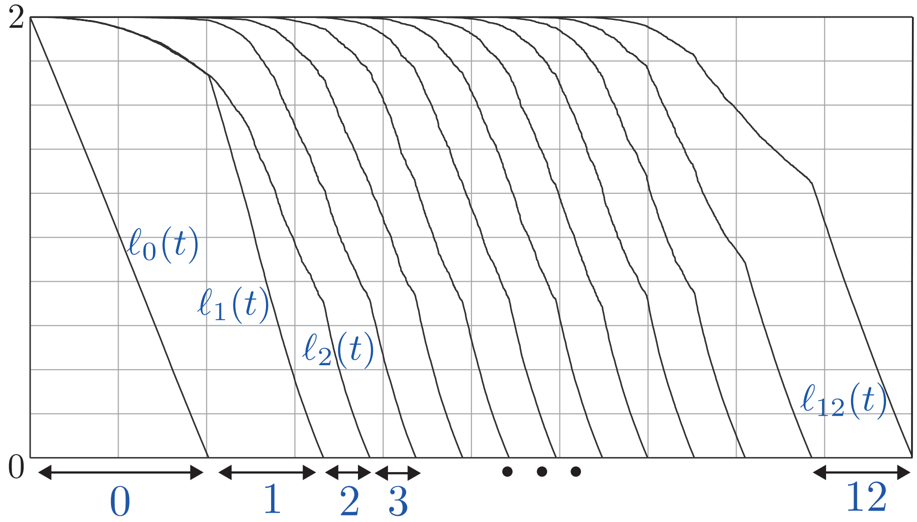

Fig. 4. A schematic representation of how the literals at each of the positions vary in time. The horizontal axis corresponds to time t which is the number of free steps. Here we have $L = 11$ and $w = 3$. This plot corresponds to an implementation of the UC algorithm on a random coupled instance. The blue numbers below the plot are the phases of the algorithm. In the beginning of the algorithm, we are in phase 0. This phase lasts until all the literals in the first position are peeled off and as a result $\ell_0(t)$ reaches 0. We then go immediately to phase 1 and this phase lasts till $\ell_1(t)$ reaches 0 and so on. We have in total $L = 11$ phases.

We now define the check types. Consider a coupled K-SAT formula to begin with. For such a formula there are $L - w + 1$ sets of check nodes placed at positions $\{0, 1, \cdots, L-1\}$. Let us consider a specific position $i \in \{0, 1, \cdots, L-1\}$ and look at the check nodes at position i. Each of these check nodes can potentially be connected to any set of K variables resting in variable positions $\{i, i+1, \cdots, i+w-1\}$. Some thought shows that there are various types of check nodes depending on the variable positions that they are connected to. For example, there is a type of check nodes for which all of the K edges go only into a single variable position $j \in \{i, i+1, \cdots, i+w-1\}$ or there is a type for with some of its edges go to position i and the rest go to position $i+1$ and so on. Also, as we proceed through the UC algorithm, some of these checks are shortened to create new types of checks with degrees less than K. We now explain a natural way to encode these various types.

By $C(t, i, \underline{\tau})$ we mean the number of check nodes at check position $i \in \{0, 1, \cdots, L-1\}$ that have type $\underline{\tau}$ at round time t. The type $\underline{\tau} = (\tau_0, \cdots, \tau_{w-1})$ is a w-tuple and indicates that relative to position i, how many edges the check has in (variable) positions $i, i+1, \cdots, i+w-1$. The best way to explain $\underline{\tau}$ is through an example. Let us assume $w = 4$ and consider the set of check nodes at check position 20 that are only connected to variable positions $20, 22, 23$ in the following way. For each of these check nodes there are exactly two edges going to position 20, and one edge going to position 22 and one edge going to position 23 (thus each of these checks have degree 4). Figure 5 illustrates a generic check node of this set. We denote the number of these checks at time t by $C(t, 20, (1, 0, 2, 1))$. In other words, the type is computed as follows. The check position number that the check rests in is 20. This check is connected to a variable at position 20, and 2 variables at position 22, and a variable at position 23. So, relative to the check position 20, we see the edge-tuple $(1, 0, 2, 1)$. Let us now repeat and generalize: By $C(t, i, \underline{\tau})$ we mean the number of check nodes, at time t, which rest in position i, and $\underline{\tau}$ is a w-tuple that indicates relative to variable position i, the number of edges that go to positions $i, i+1, \cdots, i+w-1$, respectively. One can easily see that by summing up elements of the w-tuple $\underline{\tau} = (\tau_0, \cdots, \tau_{w-1})$, we find the degree of the corresponding check type. We denote

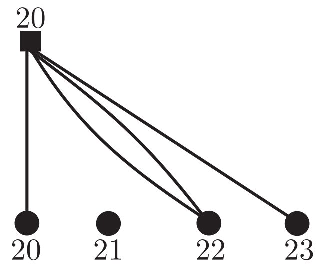

Fig. 5. A schematic representation of checks which contribute to $C(t, 20, (1, 0, 2, 1))$. All the check nodes that contribute to $C(t, i, \underline{\tau})$, were initially (at time 0) degree K check nodes resting at check position i. However, the algorithm has evolved in a way that these check nodes have been deformed (possibly shortened or remained unchanged) to have a specific type $\underline{\tau}$.

the degree of a type $\underline{\tau}$ by $\deg(\underline{\tau})$. It is also easy to see that there are $\binom{d+w-2}{d-1}$ different types of degree d for $d \in \{2, 3, \cdots, K\}$. We are now ready to write the differential equations. Our approach is as follows. Assume the phase of the algorithm is p and we are in a round t. At a free step, we fix a variable at position p (free step). This will create a number of forced steps in each of the positions $p, p+1, \cdots, L+w-2$. We first compute the average of these forced fixes in each variable position as a function of the number of degree two check nodes. Using these averages, we then update the average number of check and variable nodes at each position. We proceed by explaining a key property for the analysis.

Uniform Randomness Property The uniform randomness property means that at any round time t, for any position i and any type $\underline{\tau}$, each clause in the set $C(t, i, \underline{\tau})$ is uniformly distributed among all the possible clauses at position i with type $\underline{\tau}$. In other words, conditioned on the number of variables and check-types of different positions, the formula is uniformly random. An intuitive justification for the randomness property in our case stems from the fact that at any step (free or forced) in the UC algorithm, except the degree distribution of the remaining formula, no information what-so-ever can be deduced about the structure of the remaining formula. The exact proof of the uniform randomness property in our case can be easily deduced from [38, Lemma 3].

Inside a Round As mentioned above, a round begins with a free step and proceeds with a possible sequence of forced steps and ends when there are no more forced steps left. A crucial task in writing the differential equations is a precise characterization of the average number of forced steps taken during each round. Our objective is now to derive this average in a round t as a function of the number of degree two check nodes. In this regard, let us denote by $\beta_i(t)$ the average number of variables at position i that are set in round t. Let us also define the vector $\bar{\beta}(t)$ as

$$\bar{\beta}(t) = \begin{bmatrix} \beta_0(t) \\ \beta_1(t) \\ \vdots \\ \beta_{L+w-2}(t) \end{bmatrix}.$$

Also, it will be useful to choose a specific notation for the degree two types. For $r, s \in \{0, 1, \cdots, w-1\}$ s.t. $r < s$, we define the degree two types $\pi_{r,s}$ and $\pi_{r,r}$ as the following w-tuples

$$\pi_{r,s} = (0, \cdots, \overset{\text{position } r}{1}, \cdots, \overset{\text{position } s}{1}, \cdots, 0), \tag{33}$$

$$\pi_{r,r} = (0, \cdots, \overset{\text{position } r}{2}, \cdots, 0). \tag{34}$$

In other words, the tuple $\pi_{r,s}$ is zero at all its entries except the ones at positions r, s which it takes value one. Similarly, $\pi_{r,r}$ is non-zero only at position r where it takes value 2.

An important point to consider here is the following. Consider two variable positions $i, j \in \{0, 1, \cdots, L+w-2\}$. We ask ourselves if we set one variable at position j, how many immediate forced steps at position i would be created on average? We call this average number the *effect* of position j on i and denote it by $A_{i,j}$. To answer this, we should look at the degree two check nodes that are connected exactly to positions i and j. Let us now express $A_{i,j}$ in terms of the degree two check nodes that are connected to positions i and j. For simplicity, assume $i \leq j$. It is easy to see that if $j \geq i + w$, then $A_{i,j} = 0$. This is because each check node can only be connected to variable positions in a range of size at most w. Assuming $i + w > j$, we consider two cases: $i = j$ and $i < j$. When $i = j$, we should consider the degree two checks that are connected to position i twice. The possible positions of theses check node lie inside the set $\{i - w + 1, i - w + 2, \cdots, i\}$. For the checks at position $k \in \{i - w + 1, i - w + 2, \cdots, i\}$ the corresponding type would be $\pi_{i-k,i-k}$. Hence, we obtain

$$A_{i,i} = \frac{1}{L_i(t)} \sum_{k=i-w+1}^{i} 2C(t, k, \pi_{i-k,i-k}). \tag{35}$$

In the case where $i < j$, we need the number of checks that have one edge in position i and one in position j. So the possible check positions are inside the set $\{j - w + 1, \cdots, i\}$. For a check position k in this set, the corresponding type would be $\pi_{i-k,j-k}$. As a result, we have

$$A_{i,j} = \frac{1}{L_j(t)} \sum_{k=j-w+1}^{i} C(t, k, \pi_{i-k,j-k}). \tag{36}$$

Note that

$$L_j(t)A_{i,j} = L_i(t)A_{j,i} = \sum_{k=j-w+1}^{i} C(t, k, \pi_{i-k,j-k}). \tag{37}$$

We now define the matrix

$$A = [A_{i,j}]_{(L+w-1)\times(L+w-1)}, \tag{38}$$

that contains $A_{i,j}$ as its entries. The matrix A plays a key role in both writing the differential equations and their analysis. Assume now that we are in phase p. Having the matrix A, we can compute the vector $\bar{\beta}(t)$ via considering the multi-rate Galton-Watson tree starting at position p. We first note that in the beginning of each round we freely fix a variable at position p. This will create an effect (i.e., some forced fixes) in other positions. This effect is on average equal to Ae_p, where e_p is the vector that has a 1 in its p-th position and 0 in other positions. These new (forced) fixes will also create an effect which is on average equal to A^2e_p, and so on. Therefore, we obtain

$$\bar{\beta}(t) = (I + A + A^2 + \cdots)e_p = (I - A)^{-1}e_p. \tag{39}$$

Of course the relation (39) is valid if and only if the matrix A has spectral radius strictly less than 1. More precisely, we define the spectral radius of A as

$$\rho(A) = \max_{1\leq i\leq L+w-1} |\lambda_i|,$$

where $|\lambda_i|$ denotes the absolute value of the eigenvalue λ_i of A. For (39) to hold, we must have

$$\rho(A) < 1. \tag{40}$$

Here, a few comments are in order:

(i) We have assumed that during each round, the statistics of the formula remain constant. This condition does not completely hold, as by setting the variables the number of variables and clauses change. However, as we see in the following, by assuming $\rho(A) < 1$, the fluctuations of these statistics is $O(1)$ and when divided by L_i, their total influence would be $O(\frac{1}{N})$. As a result, they can be neglected with respect to Wormald's framework of differential equations. We thus omit such an additional factor in (39).

(ii) We notice from (37) that the matrix A can be written in the form of $A = SD$, where S is a symmetric matrix and D is a diagonal matrix. Hence, the matrix $D^{+\frac{1}{2}}AD^{-\frac{1}{2}}$ is a symmetric matrix and hence has only real eigenvalues. However, it can easily be shown that A and $D^{+\frac{1}{2}}AD^{-\frac{1}{2}}$ have the same set of eigenvalues and thus all the eigenvalues of A are real. Further with such a representation of A, one can deduce from the Perron-Frobenius formalism[6] [39] that

$$\rho(A) = \max_{1 \le i \le L+w-1} \lambda_i. \tag{41}$$

Consequently, for (39) to hold, the largest eigenvalue of A should be strictly less than 1.

The Differential Equations Now, having the vector $\bar{\beta}(t)$ we can find how the number of variables and checks evolve. For all $i \ge 0$ and all $\underline{\tau}$, let

$$\Delta L_i(t) = L_i(t+1) - L_i(t) = -2\beta_i(t),$$
$$\Delta C(t, i, \underline{\tau}) = C(t+1, i, \underline{\tau}) - C(t, i, \underline{\tau}).$$

To see how the check types evolve, we note that for a given check type there are two kinds of flows to be considered. A negative flow going out and a positive flow coming in from the checks of higher degrees. In this regard, for a type $\underline{\tau} = (\tau_0, \cdots, \tau_{w-1})$ with $\deg(\underline{\tau}) < K$ let $\partial\underline{\tau}$ be the set of types of degree $\deg(\underline{\tau}) + 1$ such that by removing one edge from them we reach to the type $\underline{\tau}$. The set $\partial\underline{\tau}$ consists of w types which we denote by $\underline{\tau}^d$, $d \in \{0, 1, \cdots, w-1\}$, such that

$$\underline{\tau}^d = \underline{\tau} + (0, \cdots, \overset{d}{1}, \cdots, 0), \tag{42}$$

where $+$ denotes vector addition in the field of reals. Thus, if $\deg(\underline{\tau}) < K$, we obtain

$$\Delta C(t, i, \underline{\tau}) = -2\sum_{d=0}^{w-1} \beta_{i+d}(t)\frac{\tau_d C(t, i, \underline{\tau})}{L_{i+d}(t)} + \sum_{d=0}^{w-1}(1+\tau_d)\beta_{i+d}(t)\frac{C(t, i, \underline{\tau}^d)}{L_{i+d}(t)}. \tag{43}$$

The right-hand side of (43) has two parts. The first part corresponds to the flow that is going out of $C(t, i, \underline{\tau})$ and has negative sign. The right part is the incoming flow from the check nodes of higher degrees. In the

[6] Every square matrix with non-negative entries, has a non-negative eigenvalue that is greater or equal in absolute value than all the other eigenvalues of the matrix.

case where $\deg(\underline{\tau}) = K$, we only have an outgoing flow since no check node with higher degrees exist. Hence, for the case $\deg(\underline{\tau}) = K$ we can write

$$\Delta C(t,i,\underline{\tau}) = -2\sum_{d=0}^{w-1} \beta_{i+d}(t)\frac{\tau_d C(t,i,\underline{\tau})}{L_{i+d}(t)}. \tag{44}$$

We now write the initial conditions for the variables and check types. Firstly, note that $L_i(0) = 2N$. In the beginning of the algorithm, all checks are of degree K, thus for types $\underline{\tau}$ such that deg($\underline{\tau}$) $< K$, we have $C(0,i,\underline{\tau}) = 0$. For deg($\underline{\tau}$) $= K$ we have

$$C(0,i,\underline{\tau}) = \alpha N\frac{\binom{K}{\tau_0,\tau_1,\cdots,\tau_{w-1}}}{w^K}. \tag{45}$$

In order to write the differential equations, we rescale the (round) time by N, i.e.

$$t \leftarrow \frac{t}{N}, \tag{46}$$

and also normalize all our other numbers by N, i.e.,

$$c(t,\cdot,\cdot) = \frac{C(Nt,\cdot,\cdot)}{N} \text{ and } \ell_i(t) = \frac{L_i(Nt)}{N}. \tag{47}$$

We then obtain for $i \in \{0,1,\cdots,L+w-2\}$,

$$\frac{d\ell_i(t)}{dt} = -2\beta_i(t). \tag{48}$$

For $i \in \{0,1,\cdots,L-w\}$ and $\deg(\underline{\tau}) < K$ we have

$$\frac{dc(t,i,\underline{\tau})}{dt} = -2\sum_{d=0}^{w-1}\beta_{i+d}(t)\frac{\tau_d c(t,i,\underline{\tau})}{\ell_{i+d}(t)} + \sum_{d=0}^{w-1}(1+\tau_d)\beta_{i+d}(t)\frac{c(t,i,\underline{\tau}^d)}{\ell_{i+d}(t)}, \tag{49}$$

and otherwise if $\deg(\underline{\tau}) = K$ we have

$$\frac{dc(t,i,\underline{\tau})}{dt} = -2\sum_{d=0}^{w-1}\beta_{i+d}(t)\frac{\tau_d c(t,i,\underline{\tau})}{\ell_{i+d}(t)}. \tag{50}$$

The vector $\bar{\beta}$ is also found as follows. For p being the current phase, we have

$$\underline{\beta}(t) = (\beta_0(t),\cdots,\beta_{L+w-2}(t))^T = (I-A)^{-1}e_p, \tag{51}$$

where $A = [A_{i,j}]_{(L+w-1)\times(L+w-1)}$ has the form

$$A_{i,j} = \frac{1}{\ell_j(t)}\begin{cases} \sum_{k=i-w+1}^{i} 2c(t,k,\pi_{i-k,i-k}) & i=j, \\ \sum_{k=j-w+1}^{i} c(t,k,\pi_{i-k,j-k}) & 0 <\mid i-j \mid< w, \\ 0 & \text{otherwise} \end{cases} \tag{52}$$

Finally, the initial conditions are given by:

$$\begin{aligned} &\ell_i(0) = 2, \text{ for } 0 \le i \le L+w-2 \\ &c(0,i,\underline{\tau}) = \begin{cases} \alpha\frac{\binom{K}{\tau_0,\tau_1,\cdots,\tau_{w-1}}}{w^K} & \text{if deg}(\underline{\tau}) = K \text{ and } 0 \le i \le L-1, \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{53}$$

The Criterion for the UC to Succeed We argue now that the criterion for the UC algorithm to succeed (i.e., to find a solution that satisfies almost all the constraints) is

$$\rho(A) \leq 1 - \delta, \tag{54}$$

for any time t, and where δ is a positive constant independent of t. We give an intuitive argument here and notice that the proof can be followed similar to [38, Lemma 4] and [32, Proposition 4.9].

Consider a particular time t and assume that the condition (54) holds. When we fix a variable at a free step, a sequence of forced steps follows. The generation of such forced variables (or unit clauses) during the round follows the pattern of a multi-rate Galton-Watson branching process. Such a process starts with a root which is the free variable that we set in the beginning of the round. Then, at every step of the process all individuals born at the previous step generate a number of offsprings. The number of offsprings in a Galton-Watson tree may follow an arbitrary fixed distribution whose mean is dependent on the position of the variables born at the previous step (the elements of the matrix A). The net result is that if $\rho(A) < 1$, then irrespective the distribution of the offsprings and their position, the population certainly becomes extinct, eventually. Mathematically speaking, this means that the Galton-Watson process is sub-critical and with probability 1, the tree is of finite size.

Assume now that we denote the Galton-Watson tree by $\mathcal{T}$. Also, with a slight abuse of notation, denote by $\mathcal{T}_i$ the set of vertices that $\mathcal{T}$ has on position i. Conditioned on $\mathcal{T}$, the probability that at position i a variable is hit more than once by $\mathcal{T}$ is $O(\frac{|\mathcal{T}_i|^2}{N})$. Hence, in expectation there are $O(\frac{|\mathbb{E}[|\mathcal{T}_i|^2]}{N})$ un-satisfied clauses that are generated as position i and at time t. Assuming $\rho(A) \leq 1 - \delta$, $\mathbb{E}[|\mathcal{T}_i|^2]$ is uniformly bounded from above by a constant for all the times t. Hence, after the UC algorithm is completed, the expected number of un-satisfied clauses at position i is $O(1)$. In fact, with a little bit work, one can show that there is a positive probability that at a position i, there are no un-satisfied clauses.

On the other hand, if $\rho(A)$ crosses the value 1 at a time t, then the corresponding Galton-Watson process becomes super-critical and it will generate with high probability a population of size $\Theta(N)$. As a result, there are $\Theta(1)$ number of clauses unsatisfied at time t. So if the value of $\rho(A)$ stays above 1 for a notable amount of time, then $\Theta(N)$ clauses would be left unsatisfied at the end of the UC algorithm.

3.4 Numerical Implementation

We have implemented the above set of differential equations (48)-(53) in C. We define the threshold $\alpha_{\mathrm{UC},L,w}(K)$ as the highest density for which the spectral norm (largest eigenvalue) of the matrix A is strictly less than one throughout the whole algorithm. A practical point to notice here is that, for the sake of implementation, we assume a phase p finishes when its corresponding variable $\ell_p(t)$ goes below a (very) small threshold $\epsilon > 0$. In our implementations, we have typically taken $\epsilon = 10^{-5}$. However, it can be made arbitrarily small as long as the computational resources allow.

Table 5 shows the value of $\alpha_{\mathrm{UC},L,w}(K)$ with $L = 50$ and $w = 3$ for different choices of K. As we observe from Table 5, for the UC algorithm with the specific schedule mentioned above, there is a significant threshold improvement over the un-coupled ensemble.

For $L = 50, w = 3, K = 3$ and several values of α, we have plotted in Figure 6 the evolution of largest eigenvalue of A as a function of round time t.

In order to characterize analytically the ultimate threshold for the UC algorithm when L and w grow large, we proceed by further analyzing the set of differential equations.

K	3	4	5
$\alpha_{\text{UC}}(K)$	2.66	4.50	7.58
$\alpha_{\text{UC},L=50,w=3}(K)$	3.67	7.81	15.76

Table 5. *First line*: The thresholds for UCP on the uncoupled ensemble. *Second line*: UCP threshold for a coupled chain with $w = 3$, $L = 50$.

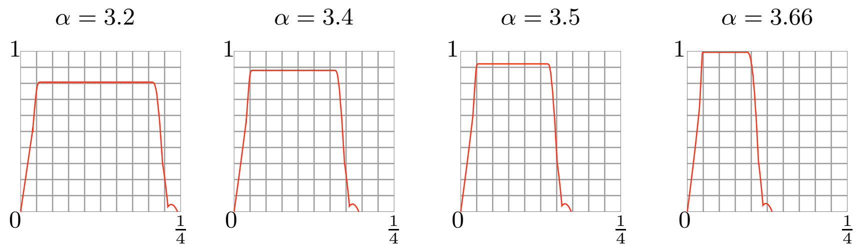

Fig. 6. The largest eigenvalue of the matrix A, plotted versus the round time t (the number of rounds divided by the total number of variables NL). The plots correspond to an actual implementation of the UC algorithm for the 3-SAT coupled ensemble with $L = 50$ and $w = 3$. As we observe, for $\alpha < 3.67$, there is a gap between the largest eigenvalue of A and the value 1 throughout the UC algorithm. By increasing α this gap shrinks to 0. For $\alpha = 3.66$ (the right-most plot) this gap is around 0.006.

3.5 Further Simplifications

The set of differential equations (48)-(53) is an autonomous system of first order differential equations with fixed initial conditions. Thus, when a solution exists, it is unique. Our objective in this section is to simplify these equations and rewrite them in a new setting that only involves the profile of literals $\{\ell_i\}_{i\geq 0}$. We note here that due to the uniqueness of the solution of these equations, our methods to simplify the equations are on the safe side.

The first basic observation is that $-2\frac{\beta_i}{\ell_i} = \frac{d}{dt}\ln \ell_i$. Therefore, the equation for $c(t, i, \underline{\tau})$ with $\deg(\underline{\tau}) < K$ can be written as

$$dc(t,i,\underline{\tau}) = \left\{\sum_{d=0}^{w-1} \tau_d (d \ln \ell_{i+d}(t))\right\} c(t,i,\underline{\tau}) - \frac{1}{2}\sum_{d=0}^{w-1}(1+\tau_d) d(\ln \ell_{i+d}(t)) c(t,i,\underline{\tau}^{(d)}), \tag{55}$$

and similarly when $\deg(\underline{\tau}) = K$ we can write

$$dc(t,i,\underline{\tau}) = \left\{\sum_{d=0}^{w-1} \tau_d (d \ln \ell_{i+d}(t))\right\} c(t,i,\underline{\tau}). \tag{56}$$

In relations (55) and (56), the term dt has been "simplified" purposedly: indeed one can view this equation as a set of first order partial differential equations. The time dependence of $c(t, i, \underline{\tau})$ is not explicit but only implicit through the $\ell_i(t)$. Therefore, in the next few lines we consider $c(t, i, \underline{\tau})$ as a function of ℓ_i's, and drop the explicit time dependence. From (56) one can see that for the case $\deg(\underline{\tau}) = K$ we have

$$\begin{aligned} \frac{\partial c(t,i,\underline{\tau})}{\partial \ln \ell_{i+d}} &= \tau_d c(t,i,\underline{\tau}),\ d = 0,\dots,w-1, \\ \frac{\partial c(t,i,\underline{\tau})}{\partial \ln \ell_{i+d}} &= 0, \qquad\qquad d \neq 0,\dots,w-1. \end{aligned} \tag{57}$$

As a result of the above relations one can easily guess that for the case $\deg(\underline{\tau}) = K$ we have

$$c(t, i, \underline{\tau}) = p_\tau \prod_{d=0}^{w-1} (\ell_{i+d})^{\tau_d}, \tag{58}$$

and also by considering the initial conditions, we obtain

$$p_\tau = \frac{\binom{K}{\tau_0, \tau_1, \cdots, \tau_{w-1}}}{w^K} \frac{\alpha}{2^K}. \tag{59}$$

We now consider types with degree less than K. It turns out that these equations can be integrated *iteratively*. In order to represent $c(t, i, \underline{\tau})$ in terms of the literals, we first need the following definition. Consider two types $\underline{\tau}$ and $\underline{\tau}'$. We say that $\underline{\tau}'$ dominates $\underline{\tau}$ if for any $d \in \{0, 1, \cdots, w-1\}$ we have $\tau_d \leq \tau'_d$. We also represent dominance by

$$\underline{\tau} \prec \underline{\tau}'. \tag{60}$$

Lemma 1. *We have for* $i \in \{0, 1, \cdots, L-1\}$

$$c(t, i, \underline{\tau}) = \prod_{d=0}^{w-1} \ell_{i+d}^{\tau_d} \sum_{\underline{\tau}' : \underline{\tau} \prec \underline{\tau}', \deg(\underline{\tau}') = K} p_{\underline{\tau}'} \prod_{d=0}^{w-1} \binom{\tau'_d}{\tau_d} (1 - \frac{\ell_{i+d}}{2})^{\tau'_d - \tau_d}. \tag{61}$$

The proof of this lemma can be done by induction and we relegate it to the appendices. For the degree 2 types, which we need in the matrix A, we find from Lemma 1, and after some simple algebra, that for $i \in \{0, 1, \cdots, L+w-2\}$ such that $k \in \{\max(i-w+1, 0), \cdots, \min(i, L-1)\}$, we have

$$c(t, k, \pi_{i-k, i-k}) = \frac{\alpha}{2^K} \frac{K(K-1)}{2w^2} \ell_i^2 \big(1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\ell_{k+d}}{2}\big)^{K-2}.$$

Also, for $i, j \in \{0, 1, \cdots, L+w-2\}$ such that $i < j$ and $k \in \{\max(j-w+1, 0), \cdots, \min(i, L-1)\}$ we have

$$c(t, k, \pi_{i-k, j-k}) = \frac{\alpha}{2^K} \frac{K(K-1)}{w^2} \ell_i \ell_j \big(1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\ell_{k+d}}{2}\big)^{K-2}.$$

This brings us to the following conclusion.

Corollary 1. *For* $i, j \in \{0, 1, \cdots, L+w-2\}$, $A_{i,j}$ *can be expressed in terms of the literals as follows. If* $0 \leq j - i \leq w-1$, *we have*

$$\begin{aligned} A_{i,j} &= \frac{\alpha}{2^K} \frac{K(K-1)}{w} \ell_i \frac{1}{w} \sum_{k=j-i}^{w-1} \Big\{ \big(1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\ell_{j-k+d}}{2}\big)^{K-2} \mathbb{1}\{0 \leq i-k \leq j-k \leq L-1\} \Big\} \\ &= \frac{\alpha}{2^K} \frac{K(K-1)}{w} \ell_i \frac{1}{w} \sum_{k=\max(j-i, j-(L-1))}^{\min(i, w-1)} \big(1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\ell_{j-k+d}}{2}\big)^{K-2}. \end{aligned}$$

Also, if $0 \leq i - j \leq w-1$, *we have*

$$A_{i,j} = \frac{\ell_i}{\ell_j} A_{j,i}, \tag{62}$$

and $A_{i,j} = 0$ otherwise. More compactly, one can write

$$A_{i,j} = \mathbb{1}\{| i - j |< w\} \frac{\alpha}{2^K} \frac{K(K-1)}{w} \ell_i \frac{1}{w} \sum_{k=\max\{|j-i|, i-(L-1), j-(L-1)\}}^{\min\{w-1,i,j\}} \big(1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\ell_{\max(i,j)-k+d}}{2}\big)^{K-2}. \tag{63}$$

Let us summarize: the differential equations can be expressed, solely in terms of the literals, as an autonomous system of first order differential equations. Assuming that we are in phase p, the differential equations take the following form

$$\frac{d\bar{\ell}}{dt} = -2(I - A)^{-1} e_p, \tag{64}$$

where the matrix A is expressed in terms of ℓ_i's as in Corollary 1 and

$$\frac{d\bar{\ell}}{dt} = \begin{bmatrix} \frac{d\ell_0}{dt} \\ \frac{d\ell_1}{dt} \\ \vdots \\ \frac{d\ell_{L+w-2}}{dt} \end{bmatrix}.$$

3.6 Conserved Quantities

The system of equations in (64) can be rewritten as

$$\sum_{j=0}^{L+w-2} (\delta_{ij} - A_{i,j}) \frac{d\ell_j}{dt} = -2\delta_{pi}, \;\; i = p, p+1, \ldots, L + w - 2, \tag{65}$$

where p denotes the phase of the algorithm. Note here that A_{ij} is given as in (63) and

$$\frac{d\ell_j}{dt} = 0, \quad \forall j \notin \{p, p+1, \cdots, L + w - 2\}. \tag{66}$$

By multiplying $\ell_i^{-1} dt$ on both sides of (65) we obtain

$$d(\ln \ell_i) - \sum_{j=i-w+1}^{i+w-1} (\ell_i^{-1} A_{i,j}) d\ell_j = -2\delta_{pi} \ell_i^{-1} dt. \tag{67}$$

Now, for positions i such that $w \le i \le L - 1$ (i.e., the position i is sufficiently away from the boundaries), by Corollary 1 the entries A_{ij} have the form

$$A_{i,j} = \mathbb{1}\{| i - j |< w\} \frac{\alpha}{2^K} \frac{K(K-1)}{w} \ell_i \frac{1}{w} \sum_{k=|j-i|}^{w-1} \big(1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\ell_{\max(i,j)-k+d}}{2}\big)^{K-2}.$$

Now, after a careful manipulation one sees that, assuming $w \le i \le L - 1$, the sum

$$\sum_{j=i-w+1}^{i+w-1} (\ell_i^{-1} A_{i,j}) d\ell_j = \frac{\alpha}{2^K} \frac{K(K-1)}{w} \sum_{j=i-w+1}^{i+w-1} d\ell_j \sum_{k=|j-i|}^{w-1} \big(1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\ell_{\max(i,j)-k+d}}{2}\big)^{K-2}$$

$$= \frac{\alpha}{2^K}\frac{K(K-1)}{w}\sum_{k=0}^{w-1}\sum_{s=0}^{w-1} d\ell_{i-k+s}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2},$$

is an exact differential form. In other words, by defining

$$Q_i \triangleq -\frac{\alpha K}{w2^{K-1}}\sum_{k=0}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-1}, \tag{68}$$

we have

$$\frac{\partial Q_i}{\partial \ell_j} = (\ell_i^{-1}A_{i,j}). \tag{69}$$

Equivalently, we can write

$$dQ_i = \sum_{j=i-w+1}^{i+w-1}(\ell_i^{-1}A_{i,j})d\ell_j. \tag{70}$$

From (75) and (67) one gets

$$d(\ln \ell_i) - dQ_i = -2\delta_{pi}\ell_i^{-1}dt. \tag{71}$$

which means that

$$P_i = \ln \ell_i - Q_i + \int_0^t 2\delta_{pi}\ell_i^{-1}dt \tag{72}$$

is a *conserved quantity* or an *integral of motion*, i.e., the value of P_i's is independent of the time t. The values of P_i's hence can be found by the initial conditions $\ell_i(0) = 2$ for $i \geq 0$ and $\ell_i(0) = 0$ for $i < 0$. Consequently, we find that for $w \leq i \leq L-1$

$$P_i = \ln 2. \tag{73}$$

However, when $i \notin \{w, \cdots, L-1\}$, i.e., when the position i is close to the boundaries, the situations is slightly different. Firstly, the quantities Q_i should be defined in accordance with the position i and secondly, given the initial profile of the algorithm, the the value of P_i can be strictly less than $\ln 2$ for some values of i. In the next section, we slightly modify our initial profile to make things more suitable for the analysis. With this initial profile, we will define the quantities Q_i for all the positions $i \in \{0, \cdots, L+w-1\}$ and also show that $P_i = \ln 2$ for all the positions i.

3.7 Slightly Modified Initial Conditions

The dependence of P_i and Q_i to the boundaries inserts some undesired asymmetry in the analysis, and makes it quiet cumbersome. As a result, we find it more convenient to slightly modify the initial conditions of the differential equations and remove such an asymmetry in the value of P_i's. As we prove in the sequel, this modification of the initial conditions does not have any effect on the final results of this paper and hence can be assumed without loss of any generality.

So, to summarize, the initial conditions

$$\ell_i(0) = \begin{cases} 0 \text{ If } i < 0, \\ 2 \text{ If } i \geq 0, \end{cases}$$

has the deficiency that the value of P_i is dependent on the position i. The objective here is to devise a new set of initial conditions on the problem so that the value of P_i given in (72) is equal to to $\ln 2$ for all the positions $i \in \{0, 1, \cdots, L+w-2\}$. In other words, we want the following relation to hold for any time t and any position i:

$$\ln \frac{\ell_i}{2} = -\frac{\alpha K}{w 2^{K-1}} \sum_{k=0}^{w-1} \big(1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\ell_{i-k+d}}{2}\big)^{K-1} \mathbb{1}\{i-k \leq L-1\} - 2 \int_0^t \delta_{pi} \ell_i^{-1} dt. \tag{74}$$

Equivalently, by defining for $i \in \{0, 1, \cdots, L+w-1\}$

$$Q_i = -\frac{K}{w 2^{K-1}} \sum_{k=0}^{w-1} \alpha_{i-k} \big(1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\ell_{i-k+d}}{2}\big)^{K-1}, \tag{75}$$

where $\alpha_{i-k} = \alpha \mathbb{1}\{i-k \leq L-1\}$, the relation (74) becomes

$$\ln \frac{\ell_i}{2} = Q_i - 2 \int_0^t \delta_{pi} \ell_i^{-1} dt. \tag{76}$$

Let us denote the new initial values for literals by $\tilde{\ell}_i$, $0 \leq i \leq L+w-2$. We further extend the profile $\{\tilde{\ell}_i\}$ to all the positions in $\mathbb{Z}$ by letting

$$\tilde{\ell}_i = 0 \;\; \forall i < 0 \tag{77}$$

$$\tilde{\ell}_i = 2 \;\; \forall i > L+w-2. \tag{78}$$

We now want to find the values $\tilde{\ell}_i$ such that initializing the differential equations with

$$\ell_i(0) = \tilde{\ell}_i, \;\; \forall i \in \mathbb{Z} \tag{79}$$

will result the relation (74). A close look at equations (74) at time $t = 0$ shows that this objective is feasible only if the profile $\{\tilde{\ell}_i\}$ is the solution of the following set of equations:

$$\ln \frac{\tilde{\ell}_i}{2} = -\frac{K}{w 2^{K-1}} \sum_{k=0}^{w-1} \alpha_{i-k} \big(1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\tilde{\ell}_{i-k+d}}{2}\big)^{K-1}, \;\; \forall i \in \{0, 1, \cdots, L+w-2\}, \tag{80}$$

where $\alpha_{i-k} = \alpha \mathbb{1}\{i-k \leq L-1\}$. In order to find a solution of (80), we can apply the iterative procedure given in Algorithm 2.

Algorithm 2 Iterative procedure to find a a solution to (80)

1: Start by initializing $\tilde{\ell}_i^0 = 2$ for $i \geq 0$ and $\tilde{\ell}_i^0 = 0$ for $i < 0$.
2: For $m = 1, 2, \cdots$ do
For $i = 0, 1, \cdots, L+w-2$ do the update

$$\tilde{\ell}_i^{m+1} = 2 \exp\{-\frac{\alpha K}{2^{K-1} w} \sum_{k=0}^{w-1} \alpha_{i-k} \big(1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\tilde{\ell}_{i-k+d}^m}{2}\big)^{K-1}\}. \tag{81}$$

Lemma 2. *The procedure of Algorithm 2 converges to a solution of the set of equations in* (80)*. We denote such a solution by* $\{\tilde{\ell}_i\}_{0 \leq i \leq L+w-2}$*. Furthermore, we have*

1. *The value of* $\tilde{\ell}_i$ *is (strictly) increasing on* $i \in \{0, 1, \cdots, L-1\}$.
2. *There exists constants* $c_1, c_2 > 0$ *such that*

$$\tilde{\ell}_i \geq 2 - c_1 e^{-c_2 (\frac{i}{w})^2}. \tag{82}$$

Proof. Consider Algorithm 2. We first show that for $m \in \mathbb{N}$ and for $i \geq 0$, we have

$$\tilde{\ell}_i^{m+1} \leq \tilde{\ell}_i^m. \tag{83}$$

With this statement in mind, it is easy to see that the profile $\{\tilde{\ell}_i^m\}_{i \geq 0}$ converges to a limit as $m \to \infty$. In order to prove (83), we first note that the function

$$h(x) = 2 \exp\{-\frac{\alpha K}{2^{K-1}}(1 - \frac{x}{2})^{K-1}\}, \tag{84}$$

is an (strictly) increasing function on the domain $x \in [0, 2]$. Using this and the definition of the profile $\{\tilde{\ell}_i^0\}_{i \in \mathbb{Z}}$, it is easy to see that for $i \in \mathbb{Z}$ we have

$$\tilde{\ell}_i^1 < \tilde{\ell}_i^0. \tag{85}$$

One can then use (85) together with (81) and the fact that h is an increasing function, to show that for $i \in \mathbb{Z}$ we have $\tilde{\ell}_i^2 < \tilde{\ell}_i^1$. Finally, by continuing this procedure inductively, we obtain (83).

Furthermore, from the fact that $h(x)$ is an increasing on $[0, 2]$ and also the fact that the profile $\{\tilde{\ell}_i^0\}_{i \in \mathbb{Z}}$ is a non-decreasing profile, we deduce that the profile $\{\tilde{\ell}_i^1\}_{i \in \mathbb{Z}}$ is also a non-decreasing profile. Again, one can generalize this statement inductively to deduce that for all $m \in \mathbb{N}$, the profiles $\{\tilde{\ell}_i^m\}_{i \in \mathbb{Z}}$ are non-decreasing profiles, and hence, the same property holds for their limit. Now, assuming that $\tilde{\ell}_i = \tilde{\ell}_{i+1}$ for some $i \in \{0, \cdots, L-1\}$, then as the profile is non-decreasing by (80) we can easily deduce that $\tilde{\ell}_{i-w+1} = \tilde{\ell}_{i-w} = \cdots = \tilde{\ell}_{i+w-1} = \tilde{\ell}_{i+w}$. But continuing in this way we obtain $\tilde{\ell}_{-1} = \tilde{\ell}_L$ and this is a contradiction as we have that $0 = \tilde{\ell}_{-1} \neq \tilde{\ell}_L = 2$. Hence, the profile is strictly increasing.

The relation (82) follows easily from Lemma 3 and we do not repeat its proof here.

As a consequence of Lemma 2, by starting with the initial conditions $\ell_i(0) = \tilde{\ell}_i$, we have $P_i = \ln 2$ for $0 \leq i \leq L - w + 2$. Also, as by (82) the profile $\{\tilde{\ell}_i\}_{i \geq 0}$ converges doubly exponentially fast in i to the value $\ell = 2$, we lose no generality in starting with $\{\tilde{\ell}_i\}_{i \geq 0}$ as our initial condition. Let us now summarize.

Theorem 3. *Starting with the initial conditions* $\{\tilde{\ell}_i\}_{i \geq 0}$*, for any time* t *during the UC algorithm, the profile of literals is a solution of the following set of equations*

$$\ln \frac{\ell_i}{2} - Q_i + \int_0^t 2\delta_{pi} \ell_i^{-1} dt = 0, \;\; \forall i \in \{0, 1, \cdots, L + w - 2\}, \tag{86}$$

where Q_i *is give in* (75)*. Furthermore, the profile of literals* $\{\ell_i(t)\}_{p \leq i \leq L-1}$ *is an (strictly) increasing profile w.r.t. the position* i*, i.e., for* $p \leq i < j \leq L-1$ *and any time* t*, we have*

$$\ell_i(t) < \ell_j(t). \tag{87}$$

The proof of this theorem is given in the appendices.

3.8 A Potential Function

From the definition of Q_i in (75), one can explicitly check the following. For $j, k \geq 0$,

$$\frac{\partial Q_k}{\partial \ell_j} = \frac{\partial Q_j}{\partial \ell_k}. \tag{88}$$

Clearly, the same statement is also true if we replace Q_i with $\ln \frac{\ell_i}{2} - Q_i$. Now, as the space of $\{\ell_i, i \geq 0\}$ is simply connected, by the Poincaré Lemma, there exists a functional Φ such that

$$\frac{\partial \Phi}{\partial \ell_i} = \ln \frac{\ell_i}{2} - Q_i,\ i \geq 0. \tag{89}$$

We call Φ the *the coupled potential function* associated to the system of differential equations or simply *the coupled potential*. Fortunately, in our case the potential Φ is easy to find. Let us first assume $L = w = 1$, i.e., the individual system. For this case we only have one literal which we denote by ℓ. From (89) and (68) we obtain

$$\frac{\partial \Phi_{\text{ind}}}{\partial \ell} = \ln \ell + \frac{\alpha K}{2^{K-1}}(1 - \frac{\ell}{2})^{K-1} - \ln 2.$$

By integrating the above relation, we get

$$\Phi_{\text{ind}}(\ell, \alpha, K) = 2 - \ell(1 - \ln \frac{\ell}{2}) - \frac{\alpha}{2^{K-2}}(1 - \frac{\ell}{2})^K, \tag{90}$$

where the constant 2 in the potential is due to the fact that, without loss of generality, we fix the potential to 0 at the point $\ell = 2$. The coupled potential can be obtained from the potential of the individual system by a simple formula derived in [17]. For our case, by using (90), we define the coupled potential to be

$$\Phi = \sum_{i=-(w-1)}^{L+w-2} 1 - \ell_i(1 - \ln \frac{\ell_i}{2}) - \frac{1}{2^{K-2}} \sum_{i=-(w-1)}^{L+w-2} \frac{1}{w} \sum_{k=0}^{w-1} \alpha_{i-k}\big(1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\ell_{i-k+d}}{2}\big)^K, \tag{91}$$

where $\alpha_{i-k} = \alpha \mathbb{1}\{i - k \leq L - 1\}$. It can easily be checked that for $i \geq 0$,

$$\frac{\partial \Phi}{\partial \ell_i} = \ln \frac{\ell_i}{2} - Q_i.$$

From now on, we call the functional Φ given in (91) the potential associated to our system.

3.9 The Threshold of the UC Algorithm for the Coupled Ensemble

Let us go back for a moment and have a look at the result of Theorem 2 in Section 2.3: for coupled systems with a state described via a one-dimensional recursion (as in (27)), the threshold of the coupled system can be computed from the potential function of the individual system by the relation (29).

An educated guess for the threshold of the UC algorithm for the coupled ensemble i the one obtained by (29), with the potential function given in (90). Figure 7 plots the value of Φ_{ind} given in (90) as a function of ℓ for different values of α and for $K = 3$. The potential threshold found in this way for $K = 3$ is 3.6717 which is extremely close to the one observed in the (numerical) solution of the differential equations (see Table 5). Is this a coincidence? Well, let us look at the case where $K = 4$. The potential threshold is found to be 7.8146 which is again very close to the one from the numerical solution of differential equations, and so on. It thus seems that the potential threshold is equal to the threshold of the UC algorithm for the coupled instances with L, w tending to infinity. This is indeed true.

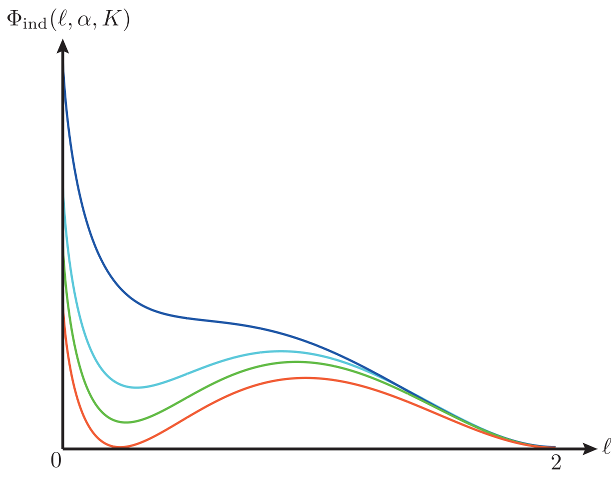

Fig. 7. The potential function associated to the individual ensemble, $\Phi_{\rm ind}(\ell, \alpha, K)$, plotted as a function of ℓ for $K = 3$ and different values of α. From top to bottom, the curves correspond respectively to $\alpha = 3.2, 3.5, 3.6, 3.67$. The smallest α for which the potential goes below the horizontal axis is around 3.67.

Theorem 4. *We have*

$$\alpha_{\rm cUC}(K) \triangleq \lim_{w\to\infty} \lim_{L\to\infty} \alpha_{{\rm UC},L,w} = \sup\{\alpha \geq 0 \mid \min_{\ell\in[0,2]} \Phi_{\rm ind}(\ell, \alpha, K) \geq 0\}. \tag{92}$$

Remark 2. For large K we find

$$\alpha_{\rm cUC}(K) \doteq 2^{K-1}. \tag{93}$$

This is roughly a factor $\frac{K}{e}$ of improvement over the threshold of the UC algorithm for the individual system that is given by

$$\alpha_{\rm UC}(K) \doteq \frac{e2^{K-1}}{K}. \tag{94}$$

However, the threshold of the coupled UC is still below the SAT/UNSAT threshold $\alpha_s(K)$ which is roughly $2^K \ln 2$. It is also below the condensation threshold which is $2^K \ln 2 - 3/2 \ln 2 + o(1)$.

The rest of this chapter is devoted to the proof of Theorem 4. We first derive several properties of the profile of the literals. We then use these properties to prove that for $\alpha \leq \alpha_{\rm cUCP}$, there exists a constant $w_0 = w(\alpha, K) < \infty$ and a constant $\delta = \delta(\alpha, K) > 0$ such that if we choose $w \geq w_0$, for any time t during the UC algorithm, we have $\rho(A) < 1 - \delta$. Hence, the fact that the UC algorithm succeeds follows from the discussions at the end of Section 3.3.

3.10 How Does the Profile Look Like?

As the first step towards the proof of Theorem 4, the objective of this section is to provide various details about the way the profile looks like during the UC algorithm. So far, the only thing that we know for sure is that given the current phase p, the profiles rests at $\ell_i = 0$ for $i < p$ and rises up to $\ell_i = 2$ for i at its right boundary, i.e., $i > L + w - 2$. Let us define the *transition region* of the profile to be the region of positions $i \in \{p, p+1, \cdots, L+w-2\}$ such that the value of ℓ_i is not very close to 2. One of the main results of this section is that the transition region is always $O(w)$ during the whole UC algorithm.

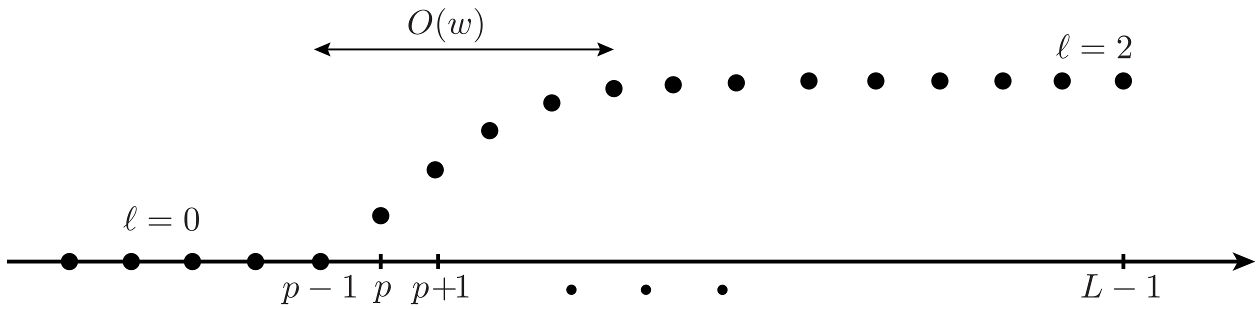

Fig. 8. A schematic representation of the profile. The transition region is of size $O(w)$.

The idea here is to consider different regions for the position $i \in \{0, 1, \cdots, L + w - 2\}$ and analyze the behavior of the profile in each of these regions. These regions are specified by the solutions the equation

$$\ln \frac{\ell}{2} = -\frac{\alpha K}{2^{K-1}}(1 - \frac{\ell}{2})^{K-1}, \tag{95}$$

or equivalently, the fixed points of the equation

$$\ell = 2 \exp\left(-\frac{\alpha K}{2^{K-1}}(1 - \frac{\ell}{2})^{K-1}\right). \tag{96}$$

The reason that we consider the fixed points of (96) stems from the conservation equations (86). Let us give an intuitive explanation. Assume w is a large but fixed number. We know that the profile is increasing. If the transition region of the profile is much larger that w (e.g., it is $O(w^2)$), then it is easy to see that there is a value $\ell^* \in (0, 2)$ such that the profile is close to ℓ^* for at least $O(w)$ positions. In other words, there is a small constant $\delta > 0$ and two positions i_1, i_2 such that $i_2 - i_1 \geq 2w$ and for $i \in \{i_1, i_1 + 1, \cdots, i_2\}$ we have $\ell_i \in [\ell^* - \delta, \ell^* + \delta]$. Now, by looking at the conservations equations (86) for a position $i = \frac{i_1+i_2}{2}$ we can easily deduce that ℓ^* should be close to a fixed point of (95).

We now proceed by specifying the solutions of (95). Let us define the function

$$f(\ell) = \ln \frac{\ell}{2} + \frac{\alpha K}{2^{K-1}}(1 - \frac{\ell}{2})^{K-1}. \tag{97}$$

The first derivative of the function f is

$$f'(\ell) = \frac{1}{\ell}\left(1 - \frac{\alpha K(K-1)}{2^K}\ell(1 - \frac{\ell}{2})^{K-2}\right). \tag{98}$$

Now, one can easily see that the equation $f'(\ell) = 0$ has at most two solutions on $\ell \in [0, 2]$. This is because the function $\ell(1 - \frac{\ell}{2})^{K-2}$ is a uni-modal function on $\ell \in [0, 2]$. Hence, by the Role Theorem, the equation $f(\ell) = 0$ (or equivalently (95)) has at most three solutions on $[0, 2]$. Indeed, a bit of calculus reveals that there exists a $\alpha^* < \alpha_{\text{cUCP}}$ such that the following holds. For $\alpha < \alpha^*$ the equation (95) has exactly one solution which is the trivial solution $\ell = 2$ and for $\alpha > \alpha^*$ there are three distinct solutions to (95). In the following, we assume the harder case, i.e., we assume that $\alpha^* < \alpha < \alpha_{\text{cUCP}}$ and hence the equation (95) has three distinct solutions on $[0, 2]$. From what we mention in the sequel, the other case, $\alpha < \alpha^*$, is much easier to analyze and in fact will follow directly from the present analysis.

As shown in Figure 9, the fixed points of (96) are obtained by intersecting the two curves

$$y_1(\ell) = 2\exp\big(-\frac{\alpha K}{2^{K-1}}(1-\frac{\ell}{2})^{K-1}\big), \tag{99}$$

$$y_2(\ell) = \ell, \tag{100}$$

on the region $\ell \in [0,2]$.

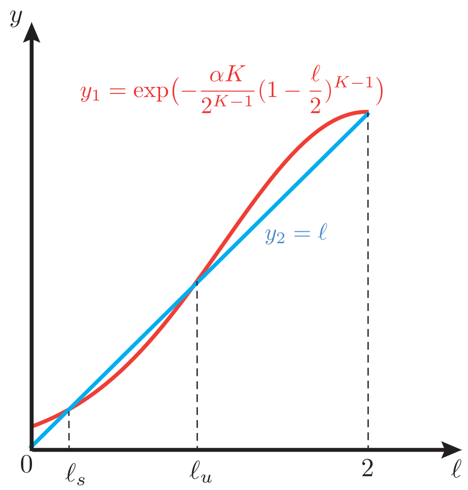

Fig. 9. A schematic representation of the fixed points of (96) which are equivalently the intersection points of the two curves $y_1(\ell)$ and $y_2(\ell)$.

As mentioned above we assume there are three distinct fixed points. The largest one $\ell = 2$ is called *the trivial fixed point*. The middle one is called *the unstable fixed point* ℓ_u and the smallest one is called *the stable fixed point* ℓ_s. Given these fixed points, we consider the following five regions for the positions $i \in \{0, 1, \cdots, L+w-2\}$. Let $\delta > 0$ be a fixed constant, the value of which will be specified in the following lemma in its suitable place. For the moment, we think of δ as a fixed and given constant. As illustrated in Figure 10, the five regions are as follows:

- region 1 (R_1): all positions i such that $\ell_i \leq \ell_s - \delta$.
- region 2 (R_2): all positions i such that $\ell_s - \delta < \ell_i \leq \ell_s + \delta$.
- region 3 (R_3): all positions i such that $\ell_s + \delta < \ell_i \leq \ell_u - \delta$.
- region 4 (R_4): all positions i such that $\ell_u - \delta < \ell_i \leq \ell_u + \delta$.
- region 5 (R_5): all positions i such that $\ell_u + \delta < \ell_i \leq 2$.

In the following lemma, we provide bounds on the behavior of the profile in each of these regions. We note here that all the results of the following lemma (and all the other results that appear later on) are only dependent on the choice of α and K, Hence, they are independent of the time or phase of the algorithm and are valid for the profile throughout the whole UC algorithm. Let us conclude this section by summarizing its results in the following lemma which is proven in the appendices.

Lemma 3. *The following properties hold for the profile of literals $\{\ell_i\}_{i \geq p}$ at any time t.*

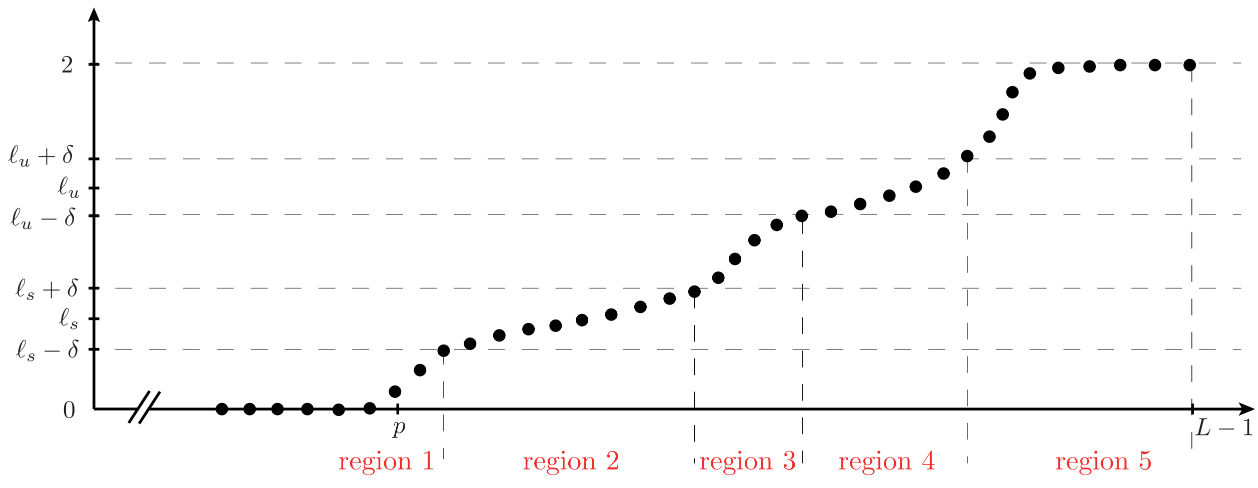

Fig. 10. Different regions for the value of the profile.

(1) For $i > p$

$$\ell_i \geq 2\exp(-\frac{\alpha K}{2^{K-1}}).$$

(2) For $i \geq p$

$$2 - \ell_{i+w} \leq \frac{\alpha K}{2^{2K-2}}(2-\ell_i)^{K-1}.$$

(3) For $i \leq L - w$

$$\ln\frac{\ell_{i+\frac{w}{2}}}{2} \leq \frac{-\alpha}{2^{K-1}}(\frac{2-\ell_i}{2})^{K-1}.$$

Also, There exist positive constants $w_0 = w_0(\alpha, K)$ and $\delta = \delta(\alpha, K)$ such that if we assume $w \geq w_0$ and define regions $R_1 - R_5$ (the regions depend on δ), then

(4) If $i \in R_1 \cup R_3$, there exists a value $\zeta = \zeta(\alpha, K) > 0$ such that $\ell_{i+w} - \ell_i \geq \zeta$. Hence, the length of the regions 1 and 3 is at most $\frac{2w}{\zeta}$.

(5) The length of the regions 2,4 is at most $2w$.

(6) Let $\epsilon > 0$ be an arbitrary positive constant. Define I_ϵ to be the first position for which the value of profile goes above $2 - \epsilon$, i.e.,

$$I_\epsilon = \text{argmin}\{j \geq p \,|\, \forall i \geq j : \;\ell_i > 2 - \epsilon\}.$$

Then, there exist constants $c_1 = c_1(\alpha, K)$ and $c_2 = c_2(\alpha, K)$ such that

$$I_\epsilon \leq p + w(c_1 + c_2 \log(\log\frac{1}{\epsilon})).$$

3.11 Why Does the UC Algorithm Work?

In this section, we take the last step in order to prove Theorem 4. That is, we show that $\rho(A)$ will be below 1 by a strict gap during the course of the UC algorithm.

Lemma 4. *There exist constants $\delta = \delta(\alpha, K) > 0$ and $w_1 = w_1(\alpha, K)$, such that if we choose $w > w_1$, then for any time t, the largest eigenvalue of matrix A is less than $1 - \delta$.*

The proof of this lemma is given in the appendices. Finally, by using (40), (41), and Lemma (4) the result of Theorem 4 follows.

References

1. S. Kudekar, T. Richardson, R. Urbanke, “Threshold saturation via spatial coupling: why convolutional LDPC ensembles perform so well over the BEC”, *IEEE Trans. Inform. Theory*, 57(2), pp.803-834, (2011).
2. S. Kudekar, T. Richardson, R. Urbanke, “Spatially coupled ensembles universally achieve capacity under belief propagation”, [online] Available: `arXiv:1201.2999 [cs.IT]`.
3. A. J. Felstrom, K. S. Zigangirov, “Time-varying periodic convolutional codes with low density parity check matrix”, *IEEE Trans. Inform. Theory*, 45 (5), pp. 2181-2190 (1999).
4. S. Kudekar and H. D. Pfister, “The effect of spatial coupling on compressive sensing,” in *Proc. of the Allerton Conf. on Communications, Control, and Computing*, Monticello, IL, USA, (2010).
5. F. Krzakala, M. Mézard, F. Sausset, Y. Sun, and L. Zdeborova, “Statistical physics-based reconstruction in compressed sensing,” [online] Available: `arXiv:1109.4424 [cs.IT]`.
6. D. Donoho, A. Javanmard, and A. Montanari, “Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing,” [online] Available: `arXiv:1112.0708 [cs.IT]`.
7. V. Chvátal and B. Reed, “Mick gets some (the odds are on his side)”, *Proc. 33rd IEEE Symp. Foundations of Computer Science (FOCS)*, pp. 620627 (1992).
8. E. Friedgut, “Sharp thresholds of graph properties, and the k-sat problem”, *J. Amer. Math. Soc.*, 12, pp. 1017-1054 (1999).
9. D. Achlioptas, E. Friedgut, “A Sharp Threshold for k-Colorability” *Random Structures & Algorithms*, 14(1), pp. 63-70 (1999).
10. D. Gamarnik, “Linear phase transition in random linear constraint satisfaction problems”, *Prob. Th. & Rel. Fields*, 129 (3), pp. 410440 (2004).
11. M. Bayati, D. Gamarnik, and P. Tetali, “Combinatorial approach to the interpolation method and scaling limits in sparse random graphs”, *in proc. STOC*, pp. 105-114, (2010).
12. L. Zdeborova, F. Krzakala, “Phase transitions in the coloring of random graphs”, *Phys. Rev. E*, vol 76, 031131, (2007).
13. O. Dubois and J. Mandler, “The 3-XORSAT Threshold”, *In proc. of 43rd IEEE FOCS* pp.769-778, (2002)
14. S. H. Hassani, N. Macris, R. Urbanke, “Threshold saturation in spatially coupled constraint satisfaction problems”, *J. Stat. Phys.: theory and experiments*, (2012).
15. F. Krzakala, A. Montanari, F. Ricci-Tersenghi, G. Semerjian, L. Zdeborova, “Gibbs states and the set of solutions of random constraint satisfaction problems”, *Proc. National Academy of Sciences*, vol 104, pp. 10318-10323, (2007).
16. M. Mézard, F. Ricci-Tersenghi, R. Zecchina, “Alternative Solutions to Diluted p-spin Models and XORSAT Problems”, *J. Stat. Phys*, vol 111, pp.505-533 (2003)
17. A. Yedla, Y. Jian, P. S. Nguyen and H. D. Pfister, “A simple proof of threshold saturation for coupled scalar recursions,” *in proc. ISTC*, Gothenburg, Sweden (2011).
18. S. Kudekar, T. Richardson and R. Urbanke,“Wave-like solutions of general one-dimensional spatially coupled systems,” [online] Available: `arXiv:1208.5273 [cs.IT]`.
19. S. H. Hassani, N. Macris, R. Urbanke, “Coupled graphical models and their threshold”, *Information Theory Workshop (ITW)*, Dublin, (2010); also in `lanl.arxiv.org no 1105.0785[cs.IT]`
20. S. H. Hassani, N. Macris, R. Urbanke, “Chains of mean field models”, *Journal of Statistical Physics: Theory and Experiments*, (2012).
21. D. Achlioptas and Y. Peres, “The threshold for random K-SAT is $2^K \log 2 - O(K)$”, *J. Amer. Math. Soc.*, 17(4), pp. 947973, (2004).
22. M. T. Hajiaghayi and G. B. Sorkin, “The satisfiability threshold of random 3-SAT is at least 3.52”, *volume RC22942 of IBM Research Report*, (2003).
23. A. C. Kaporis, L. M. Kirousis, and E. G. Lalas, “The probabilistic analysis of a greedy satisfiability algorithm”, *Random Structures and Algorithms*, 28(4), pp. 444480, (2006).
24. O. Dubois, Y. Boufkhad, and J. Mandler,“Typical random 3-SAT formulae and the satisfiability threshold”, *Electronic Colloquium on Computational Complexity (ECCC)*, 10(007), (2003).
25. O. Dubois, and Y. Boufkhad , “A general upper bound for the satisfiability threshold of random r-SAT formulae”, *J. Algorithms*, 24(2), pp. 395-420, (1997).
26. L. M. Kirousis, E. Kranakis, D. Krizanc, and Y. Stamatiou, “Approximating the unsatisfiability threshold of random formulas”, *Random Structures Algorithms*, 12(3), pp. 253-269, (1998).
27. M. Mézard, M. Palassini and O. Rivoire, “Landscape of solutions in constraint satisfaction problems”, *Phys. Rev. Lett.*, vol 95, pp. 200-202, (2005).
28. D. Achlioptas and Y. Peres, “The threshold for random K-SAT is $2^K \log 2 - O(K)$”, *J. Amer. Math. Soc.*, 17(4), pp. 947973, (2004).
29. D. Achlioptas and C. Moore, “Random K-SAT: Two moments suffice to cross a sharp threshold”, *SIAM J. Comput.*, 36(3), pp. 740762, (2006).

30. A. Coja-Oghlan and K. Panagiotou, “Going after the K-SAT threshold”, [online] Available: `arXiv:1212.1682 [math.CO]`.
31. M. T. Hajiaghayi and G. B. Sorkin, “The satisfiability threshold of random 3-SAT is at least 3.52”, *volume RC22942 of IBM Research Report*, (2003).
32. A. C. Kaporis, L. M. Kirousis, and E. G. Lalas, “The probabilistic analysis of a greedy satisfiability algorithm”, *Random Structures and Algorithms*, 28(4), pp. 444480, (2006).
33. O. Dubois, Y. Boufkhad, and J. Mandler,“Typical random 3-SAT formulae and the satisfiability threshold”, *Electronic Colloquium on Computational Complexity (ECCC)*, 10(007), (2003).
34. D. Achlioptas, C. Moore, “All graphs with average degree 4 are 3-colorable,” *Journal of Computer and System Sciences 67*, pp. 441-471, (2003) .
35. D. Achlioptas, A. Noar, “The two possible values of the chromatic number of a random graph,” *Annals of Mathematics 162*, pp. 1335-1351, (2005).
36. A. Coja-Oghlan, D. Vilenchik, “Chasing the k-colorability threshold,” *In proc. 54th FOCS*, pp. 380-389, (2013).
37. B. Pittel, J. Spencer, and N. Wormald, “Sudden emergence of a giant k-core in a random graph”, *J. Comb. Theory. B*, vol 67, pp. 111-151, (1996).
38. D. Achlioptas, “Lower bounds for random 3-SAT via differential equations,” *Theoretical Computer Science 265*, pp. 159-185, (2001) .
39. C. Godsil and G. Royle, *Algebraic Graph Theory*, Springer, (2001).

A Auxiliary Lemmas and Proofs

A.1 A Message Passing Interpretation For UC

In this section we aim to express the UC algorithm in a message passing (MP) formalism. We start by explaining the MP formalism on the individual ensemble (i.e., $L = w = 1$). For this ensemble we will analyze the dynamics of the MP procedure in a probabilistic manner and derive the so called density evolution (DE) equations of the MP procedure. We then extend the MP procedure and the DE equations to the coupled ensemble. Let us stress again the fact that the following MP procedure is designed to formulate the dynamics of the UC algorithm in a message passing fashion.

We proceed by recalling that a formula in the individual ensemble can be thought as a bipartite graph with N literals (variable nodes) and $M = N\alpha$ clauses (check nodes). We denote a check by $c, h \in \{0, 1, \cdots, M-1\}$ and the variables by $i, j \in \{0, \cdots, N-1\}$. Each check has K edges which randomly have chosen one of the N variables. So the graph has MK edges. An edge of the graph, between a check node c and a variable node i is also denoted by $\langle c, i\rangle$. Also, each edge has an associated sign, being -1 or $+1$ with equal probability, which we denote by $J_{c,i}$.

For each edge (c, i) of the graph we associate two types of messages: (i) the check-to-variable message $\mu_{c\to i}$ which takes its value in the set $\{0, 1\}$ (ii) the variable-to-check message, $(\mu_{i\to c}, s_{i\to c})$ which the value of $\mu_{i\to c}$ is inside $\{0, 1\}$ and the value of $s_{i\to c}$ is in $\{-1, 1\}$. On the intuitive level, these messages are designed to mimic the behavior of the UC algorithm. In this regard, when a message $\mu_{c\to i}$ is 1, this means that the check c is forcing the variable i to satisfy it. This situation occurs in the course of the UC algorithm when the check c is a unit clause. Furthermore, when a message $\mu_{i\to c}$ takes the value 1, this means that the variable i tells the check c that it has a preset value. The other message $s_{i\to c}$ is the preset value of variable i that it sends to check c.

The MP procedure consists of N steps. At each step $r = 1, \cdots, N$ we choose a variable, *reveal* it, and update the messages. Thus, in the end of MP all the variables are revealed. We now describe the MP procedure in detail through the following stages.

Initialization: In the beginning of the MP procedure, we initialize all the messages $\mu_{c\to i}$ and $\mu_{i\to c}$ to 0. This indicates that in the beginning all the checks and variables tell each other that they are essentially free. Let s_i be a bernoulli rv whose value is chosen by flipping a fair coin. By flipping a fair coin we specify the value of s_i and give this value to all the messages $s^0_{i\to c}$. In other words, we start from a random assignment on all the variables (and as the algorithm proceeds, this assignment changes by the presense of unit-clauses).

During step r: In each step $r \in \{1, \cdots, N\}$, we first choose one variable uniformly at random among the remaining unrevealed variables. Let us denote the chosen variable by i. We do the following operations once i is chosen.

1. We mark the variable i as revealed.
2. We then fix the value of all the outgoing messages $\mu_{i\to c}$ to 1 for all $c \in \partial i$.
3. Finally, we update the messages as follows. At each step r, we consider the messages passed from the previous step (step $r-1$), and we run the following update rules until we reach a fixed state on the messages and no further updates is necessary. We then pass these final messages to the next step (step $r+1$). Consider an edge $\langle c, i\rangle$. The check to variable message we have

$$\mu^{u+1}_{c\to i} = \prod_{j\in\partial c\setminus i} \mathbb{1}\{\mu^u_{j\to c} = 1, s^u_{j\to c} = J_{c,j}\}. \tag{101}$$

For the variable to check messages on $\langle c, i\rangle$ we do

$$\mu^{u+1}_{i\to c} = 1 - \prod_{h\in\partial i\setminus c} \mathbb{1}\{\mu^u_{h\to i} = 0\}, \tag{102}$$

$$s^{u+1}_{i\to c} = s^u_{i\to c}\mathbb{1}\{\mu^{u+1}_{i\to c} = 0\} + (-1)^{\frac{1+J_{c,i}}{2}}\mathbb{1}\{\mu^{u+1}_{i\to c} = 1\}. \tag{103}$$

Let us now analyze the MP procedure by the method of density evolution. When we reveal a variable, the variable is either "known" a priori as a consequence of some unit-clause steps, or it is unknown and we learn its value upon its relevance. We call the former case a trivial reveal and we call the latter a non-trivial reveal. Let us now define the necessary notation for the analysis. Assume that we are at the r-th step of the algorithm. We let

– s: the current step of the algorithm divided by N (i.e. $s = \frac{r}{N}$) .
– t: the current time of the algorithm which is defined as the fraction of non-trivial reveals up to now (up to the r-th step) divided by N.
– $x(t)$: fraction of the variable to check messages whose value is 0.
– $y(t)$: fraction of the check to variable messages whose value is 0.
– $k(t)$: the fraction of the variables which are either revealed by now or whose value is known as a consequence of some unit-clause steps.
– $\ell(t)$: the fraction of literals whose value is unknown divided by N, i.e. $\ell(t) = 2(1 - k(t))$.

By using the update rule (101), we have for any time t

$$y(t) = 1 - (\frac{1 - x(t)}{2})^{K-1}. \tag{104}$$

Also, from (102) we obtain

$$x(t) = (1 - s)\exp\bigl(-\alpha K(1 - y(t))\bigr). \tag{105}$$

Also, since the degree distribution of the variable nodes is Poisson, it is easy to see that the fraction of unknown variables, $1 - k(t)$, obeys the same update equation as for $x(t)$, and we thus have

$$1 - k(t) = x(t), \tag{106}$$

or equivalently

$$\ell(t) = 2x(t). \tag{107}$$

One can also relate s, t and $k(t)$ via the following simple argument. Assume that we are in the beginning of step r and we want to reveal a variable. If randomly choose a variable among the $(1-s)N$ variables that are still left, then with probability $\frac{1-k(t)}{1-s}$ this variable is an unknown variable. As a result, we have

$$dt = \frac{1-k(t)}{1-s} ds, \tag{108}$$

and by integrating both sides we obtain

$$\ln(1-s) = -\int_0^t \frac{dt}{1-k(t)}. \tag{109}$$

Finally, by combining the relations (104)-(109), we obtain

$$\ln \frac{\ell(t)}{2} = -\int_0^t \frac{2}{\ell(t)} dt - \frac{\alpha K}{2^{K-1}} (\frac{1-\ell(t)}{2})^{K-1}. \tag{110}$$

A.2 Proof of Lemma 1

We prove the lemma by induction on the degree of the type $\underline{\tau}$. Assuming deg$(\underline{\tau}) = K$, the result follows trivially from (58). Now, by assuming that (61) is true from all the types of degree not smaller than an integer q, we show that (61) holds also for all the types of degree $q-1$. To this end, it is enough to show that for the candidate of $c(t, i, \underline{\tau})$ given in (61) such that deg$(\underline{\tau}) = q-1$, the relation (55) (or equivalently (49)) holds. Rewriting (61), we have

$$c(t,i,\underline{\tau}) = \underbrace{\prod_{d=0}^{w-1} \ell_{i+d}^{\tau_d}}_{b_1} \underbrace{\sum_{\underline{\tau}': \underline{\tau} \prec \underline{\tau}', \deg(\underline{\tau}')=K} p_{\underline{\tau}'} \prod_{d=0}^{w-1} \binom{\tau'_d}{\tau_d} (1 - \frac{\ell_{i+d}}{2})^{\tau'_d - \tau_d}}_{b_2},$$

and hence

$$dc(t, i, \underline{\tau}) = b_1 \times db_2 + b_2 \times db_1. \tag{111}$$

We can then write

$$\begin{aligned} b_2 \times db_1 &= b_2 \times \prod_{d=0}^{w-1} \ell_{i+d}^{\tau_d} \sum_{j=0}^{w-1} \tau_j \frac{d\ell_{i+j}}{\ell_{i+j}} \\ &= b_2 \times b_1 \sum_{j=0}^{w-1} \tau_j (d \ln \ell_{i+j}(t)) \\ &= c(t, i, \underline{\tau}) \sum_{j=0}^{w-1} \tau_j (d \ln \ell_{i+j}(t)), \end{aligned} \tag{112}$$

and

$b_1 \times db_2$

$$
\begin{aligned}
&= -\frac{1}{2}b_1 \times \sum_{j=0}^{w-1} d\ell_{i+j} \sum_{\underline{\tau}':\,\underline{\tau}\prec\underline{\tau}',\deg(\underline{\tau}')=K} p_{\underline{\tau}'} \binom{\tau_j'}{\tau_j}(\tau_j'-\tau_j)(1-\frac{\ell_{i+j}}{2})^{\tau_j'-\tau_j-1} \prod_{d=0,d\neq j}^{w-1} \binom{\tau_d'}{\tau_d}(1-\frac{\ell_{i+d}}{2})^{\tau_d'-\tau_d} \\
&= -\frac{1}{2}b_1 \times \sum_{j=0}^{w-1} d\ell_{i+j} \sum_{\underline{\tau}':\,\underline{\tau}\prec\underline{\tau}',\deg(\underline{\tau}')=K} p_{\underline{\tau}'} \binom{\tau_j'}{1+\tau_j}(1+\tau_j)(1-\frac{\ell_{i+j}}{2})^{\tau_j'-\tau_j-1} \prod_{d=0,d\neq j}^{w-1} \binom{\tau_d'}{\tau_d}(1-\frac{\ell_{i+d}}{2})^{\tau_d'-\tau_d} \\
&= -\frac{1}{2}b_1 \times \sum_{j=0}^{w-1} d\ell_{i+j}(1+\tau_j) \sum_{\underline{\tau}':\,\underline{\tau}\prec\underline{\tau}',\deg(\underline{\tau}')=K} p_{\underline{\tau}'} \binom{\tau_j'}{1+\tau_j}(1-\frac{\ell_{i+j}}{2})^{\tau_j'-\tau_j-1} \prod_{d=0,d\neq j}^{w-1} \binom{\tau_d'}{\tau_d}(1-\frac{\ell_{i+d}}{2})^{\tau_d'-\tau_d} \\
&= -\frac{1}{2}b_1 \times \sum_{j=0}^{w-1} d\ell_{i+j}(1+\tau_j) \frac{c(t,i,\underline{\tau}^{(j)})}{\ell_{i+j}\prod_{d=0}^{w-1}\ell_{i+d}^{\tau_d}} \\
&\overset{(a)}{=} -\frac{1}{2}b_1 \times \sum_{j=0}^{w-1} d\ell_{i+j}(1+\tau_j) \frac{c(t,i,\underline{\tau}^{(j)})}{\ell_{i+j}b_1} \\
&= -\frac{1}{2}\sum_{j=0}^{w-1} d\ln(\ell_{i+j})(1+\tau_j)c(t,i,\underline{\tau}^{(j)}).
\end{aligned} \tag{113}
$$

Here, step (a) follows from the the assumption of the induction that (61) holds for all the types with degree at least q. Now, by (111), (112), and (113) we can conclude that (61) is also true for all the types with degree $q-1$.

A.3 Proof of Theorem 3

The proofs of Theorem 3 and Lemma 2 share several similar arguments. Therefore, we advise the reader to go through the proof of Lemma 2 before the proof given here. The relation (86) follows from the discussions given before the statement of Theorem 3. What remains to be proven is that the profile is increasing for any time t. We first show that the profile is non-decreasing. Define

$$
r_t = \exp\{-\int_0^t 2\delta_{pi}\ell_i^{-1}dt\}, \tag{114}
$$

and note that r_t is non-increasing by t. Consider the following the recursion of Algorithm 3. Note here that, by assuming the profile $\{\ell_i(t)\}_{i\geq 0}$ is non-increasing at time t, the profile that we obtain at every step m of Algorithm 3 is a non-decreasing profile. The reason for this is similar to the one given in the proof of Lemma 2 and hence we omit it. We now show that this recursion of Algorithm 3 converges to $\{\ell_i(t+dt)\}_{i\geq 0}$ and hence we can conclude that the profile is also non-decreasing at time $t+dt$.

Consider the initial profile $\{\tilde{\ell}_i^0\}_{i\geq 0}$. Clearly this profile is non-decreasing as $\{\ell_i(t)\}_{i\geq 0}$ is non-decreasing. One can use a similar argument as given in the proof of Lemma 80 to show that the relation (83) is also valid for the recursion of Algorithm 3. Therefore, as $m\to\infty$, the output profile of Algorithm 117 converges to a fixed point of (116)-(117). We denote this final profile by $\tilde{\ell}_i^\infty$, and our objective for the rest of the proof is to show that the profiles $\{\tilde{\ell}_i^\infty\}_{i\geq 0}$ and $\{\ell_i(t+dt)\}_{i\geq 0}$ are indeed the same.

By using the relations (114), (116)-(117), and (63), one can write

$$
\begin{bmatrix} \tilde{\ell}_p^1 \\ \tilde{\ell}_{p+1}^1 \\ \vdots \\ \tilde{\ell}_{L-1}^1 \end{bmatrix} - \begin{bmatrix} \ell_p(t) \\ \ell_{p+1}(t) \\ \vdots \\ \ell_{L-1}(t) \end{bmatrix} = -2dt(I+A)e_p + O((dt)^2) \tag{118}
$$

Algorithm 3

1: Start by initializing

$$\tilde{\ell}_i^0 = \begin{cases} \ell_p(t) - 2dt & i = p, \\ \ell_i(t) & i \neq p \end{cases} \tag{115}$$

2: For $m = 1, 2, \cdots$ do

For $i = p$ do the update

$$\tilde{\ell}_p^{m+1} = 2r_{t+dt} \exp\{-\frac{K}{2^{K-1}w} \sum_{k=0}^{w-1} \alpha_{i-k} \big(1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\tilde{\ell}_{i-k+d}^m}{2}\big)^{K-1}\}. \tag{116}$$

For $i = p+1, 1, \cdots, L-1$ do the update

$$\tilde{\ell}_i^{m+1} = 2 \exp\{-\frac{K}{2^{K-1}w} \sum_{k=0}^{w-1} \alpha_{i-k} \big(1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\tilde{\ell}_{i-k+d}^m}{2}\big)^{K-1}\}. \tag{117}$$

Continuing this way, we have for $m > 1$

$$\begin{bmatrix} \tilde{\ell}_p^m \\ \tilde{\ell}_p^m \\ \vdots \\ \tilde{\ell}_{L-1}^m \end{bmatrix} - \begin{bmatrix} \ell_p(t) \\ \ell_{p+1}(t) \\ \vdots \\ \ell_{L-1}(t) \end{bmatrix} = -2dt(I + A + A^2 + \cdots + A^m)e_p + O((dt)^2), \tag{119}$$

and as a result, provided that (54) holds, we obtain

$$\begin{bmatrix} \tilde{\ell}_p^\infty \\ \tilde{\ell}_{p+1}^\infty \\ \vdots \\ \tilde{\ell}_{L-1}^\infty \end{bmatrix} - \begin{bmatrix} \ell_p(t) \\ \ell_{p+1}(t) \\ \vdots \\ \ell_{L-1}(t) \end{bmatrix} = -(I + A + A^2 + \cdots)e_p dt = 2(I - A)^{-1} e_p dt + O((dt)^2).$$

This relation is exactly equivalent to (64). Hence, we can conclude that the profiles $\{\tilde{\ell}_i^\infty\}_{i\geq 0}$ and $\{\ell_i(t + dt)\}_{i\geq 0}$ are the same.

A.4 Proof of Lemma 3

Proof of part (1): we start by a key observation. At a phase p, the conservation equations (86) for a position $i > p$ reads

$$\ln \frac{\ell_i}{2} = -\frac{K}{w2^{K-1}} \sum_{k=0}^{w-1} \alpha_{i-k} \big(1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\ell_{i-k+d}}{2}\big)^{K-1}, \ \ \forall i > p, \tag{120}$$

or equivalently,

$$\ell_i = 2 \exp\{-\frac{K}{w2^{K-1}} \sum_{k=0}^{w-1} \alpha_{i-k} \big(1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\ell_{i-k+d}}{2}\big)^{K-1}\}, \ \ \forall i > p. \tag{121}$$

It is easy to see that for $i \in \{p+1, \cdots, L-1\}$, the relation (121) can be written in the form of

$$\ell_i = f(\frac{1}{w} \sum_{k=0}^{w-1} g(\frac{1}{w} \sum_{d=0}^{w-1} \ell_{i-k+d})), \tag{122}$$

where the functions f and g are given as

$$f(x) = 2\exp(\frac{\alpha K}{2^{K-1}}x), \tag{123}$$

$$g(x) = -(1-\frac{x}{2})^{K-1}. \tag{124}$$

Also, for $i = p$ we have

$$\ell_p \leq f(\frac{1}{w}\sum_{k=0}^{w-1} g(\frac{1}{w}\sum_{d=0}^{w-1}\ell_{p-k+d})), \tag{125}$$

and for $i > L-1$ we have

$$\ell_i \geq f(\frac{1}{w}\sum_{k=0}^{w-1} g(\frac{1}{w}\sum_{d=0}^{w-1}\ell_{i-k+d})). \tag{126}$$

Consider a position $i > p$. By using (122) and (126) we can write

$$\begin{aligned}\ell_i &\geq f(\frac{1}{w}\sum_{k=0}^{w-1} g(\frac{1}{w}\sum_{d=0}^{w-1}\ell_{i-k+d}))\\ &\stackrel{(a)}{\geq} f(\frac{1}{w}\sum_{k=0}^{w-1} g(\frac{1}{w}\sum_{d=0}^{w-1}\ell_{i-w}))\\ &= f(g(\ell_{i-w})).\end{aligned}$$

Here, step (a) follows from the fact that the profile of literals is an increasing profile and also from the fact that the functions f and g are increasing functions on the unit interval.

Using a similar argument, one can also deduce that for $i \in \{p, \cdots, L-1\}$, we have $\ell_i \leq f(g(\ell_{i+w}))$. Also, for $i \geq L$ we know by definition that $\ell_{i+w} = 2$, and thus we clearly have for $i \geq L$ that $\ell_i \leq f(g(\ell_{i+w}))$. So to summarize, we have

$$\ell_i \geq f(g(\ell_{i-w})) \;\; \forall i > p, \tag{127}$$

$$\ell_i \leq f(g(\ell_{i+w})), \;\; \forall i \geq p. \tag{128}$$

The first consequence of (127) is that

$$\ell_i \geq f(g(0)) = 2\exp(-\frac{\alpha K}{2^{K-1}}), \;\; \forall i > p. \tag{129}$$

Hence, part (1) of the lemma is proved.

Proof of part (2): By (127) we have for $i \geq p$

$$\begin{aligned}\ell_{i+w} &\geq f(g(\ell_i))\\ &= 2\exp\big(-\frac{\alpha K}{2^{K-1}}(1-\frac{\ell_i}{2})^{K-1}\big)\\ &\geq 2(1-\frac{\alpha K}{2^{K-1}}(1-\frac{\ell_i}{2})^{K-1}).\end{aligned}$$

Thus, by rearranging terms we get for $i \geq p$

$$2-\ell_{i+w} \leq \frac{\alpha K}{2^{2K-2}}(2-\ell_i)^{K-1}.$$

Proof of part (3): starting from the conservation equations (86), we have for $i \leq L - w$

$$
\begin{aligned}
\ln \frac{\ell_{i+\frac{w}{2}}}{2} &= -\frac{\alpha K}{2^{K-1}} \frac{1}{w} \sum_{k=0}^{w-1} (\frac{1}{w} \sum_{d=0}^{w-1} \frac{2 - \ell_{i+\frac{w}{2}+d-k}}{2})^{K-1} \\
&\leq -\frac{\alpha K}{2^{K-1}} \frac{1}{w} \sum_{k=\frac{w}{2}}^{w-1} (\frac{1}{w} \sum_{d=0}^{w-1} \frac{2 - \ell_{i+\frac{w}{2}+d-k}}{2})^{K-1} \\
&\leq -\frac{\alpha K}{2^{K-1}} \frac{1}{w} \sum_{s=0}^{\frac{w}{2}-1} (\frac{1}{w} \sum_{d=0}^{w-1} \frac{2 - \ell_{i+d-s}}{2})^{K-1} \\
&\leq -\frac{\alpha K}{2^{K-1}} \frac{1}{w} \sum_{s=0}^{\frac{w}{2}-1} (\frac{1}{w} \sum_{d=0}^{s} \frac{2 - \ell_{i+d-s}}{2})^{K-1} \\
&\leq -\frac{\alpha K}{2^{K-1}} \frac{1}{w} \sum_{s=0}^{\frac{w}{2}-1} (\frac{1}{w} \sum_{d=0}^{s} \frac{2 - \ell_{i}}{2})^{K-1} \\
&\leq -\frac{\alpha K}{2^{K-1}} \frac{1}{w} \sum_{s=0}^{\frac{w}{2}-1} (\frac{s+1}{w} \frac{2 - \ell_{i}}{2})^{K-1} \\
&\leq -\frac{\alpha K}{2^{K-1}} (\frac{2 - \ell_{i}}{2})^{K-1} \frac{1}{w} \sum_{s=1}^{\frac{w}{2}} (\frac{s}{w})^{K-1} \\
&\leq -\frac{\alpha K}{2^{K-1}} (\frac{2 - \ell_{i}}{2})^{K-1} \int_0^{\frac{1}{2}} x^{K-1} dx \\
&= -\frac{\alpha}{2^{2K-1}} (\frac{2 - \ell_{i}}{2})^{K-1}.
\end{aligned}
$$

Proof of part (4): we first consider the region 1 (R_1). In this regard, we define the sequence $\{x_j\}_{j \in \mathbb{N}}$ with $x_0 = 0$ and for $j \geq 0$

$$x_{j+1} = f(g(x_j)). \tag{130}$$

Now, by using the fact that for $i \geq p$ we have $\ell_{i+w} \geq f(g(\ell_i))$, we obtain

$$\ell_{p+jw} \geq x_j, \ \ \forall j \geq 0. \tag{131}$$

We now claim that there exist a positive value $m_s \in (0, 1)$ such that for $j \geq 0$

$$\ell_s - x_{j+1} \leq m_1(\ell_s - x_j). \tag{132}$$

This claim can easily be deduced from Figure 11. A little calculus reveals that the line tangent to the curve $y_1(\ell) = f(g(\ell))$ at the point $\ell = \ell_s$ stays below the curve $y_1(\ell)$ when $\ell \in [0, \ell_s]$ (see Figure 11). Thus, by defining

$$m_1 = y_1'(\ell_s) < 1, \tag{133}$$

we have for $\ell \leq \ell_s$

$$y_1(\ell) \geq \ell_s - m_1(\ell_s - \ell). \tag{134}$$

Hence, one can write for $j \geq 0$

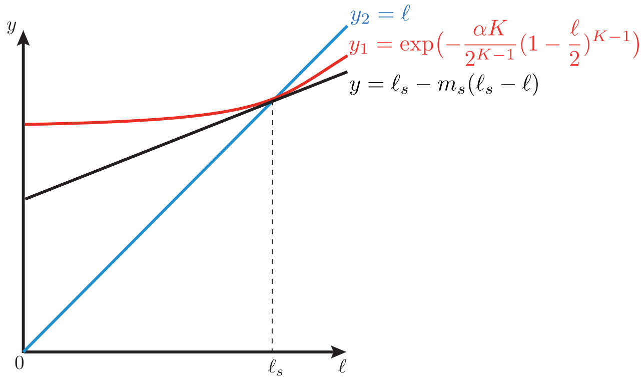

Fig. 11. The line $y(\ell)$ is the tangent line to the curve $y_1(\ell)$ at the fixed point $\ell = \ell_s$. As we see from the figure, for $\ell \in [0, \ell]$, the line $y(\ell)$ stays between the curves $y_1(\ell)$ and $y_2(\ell)$.

$$x_{j+1} = y_1(x_j) \geq \ell_s - m_1(\ell_s - x_j),$$

and as a result

$$\ell_s - x_{j+1} \leq m_1(\ell_s - x_j),$$

Thus, the sequence $\{x_j\}_{j\in\mathbb{N}}$ converges exponentially fast to ℓ_s.

Another important consequence of the discussion above is that for a position $i \in R_1$ we have

$$\begin{aligned} \ell_{i+w} &\geq f(g(\ell_i)) \\ &\overset{(a)}{\geq} \ell_s - m_1(\ell_s - \ell_i) \\ &= (1 - m_1)\ell_s + m_1\ell_i \\ &\overset{(b)}{\geq} \delta(1 - m_1) + \ell_i, \end{aligned}$$

where step (a) follows from (134) and (b) follows from the fact that $\ell_i \leq \ell_s - \delta$. As a result, for $i \in R_1$ we have that

$$\ell_{i+w} - \ell_i \geq (1 - m_1)\delta. \tag{135}$$

For region 3, one can use the same ideas (as in region 1) but with a bit of more effort to justify that the same results as in region 1 hold here. That is, for $i \in R_3$ we have

$$\ell_{i+w} - \ell_i \geq (1 - m_3)\delta, \tag{136}$$

for a positive constant $m_3 < 1$.

Now, by letting

$$\zeta = \delta \min(1 - m_1, 1 - m_3), \tag{137}$$

the proof of part (4) follows from (135) and (136). We further notice that, as we will specify shortly, the value of δ is chosen to be merely dependent on α and K.

Proof of part (5): the proof for regions 2 and 4 requires different techniques than the ones presented above. In fact, one can note that in the above justifications we did not use the fact that $\alpha < \alpha_{\text{cUCP}}$. Here, we use this assumption. We also adjust the value of δ in a suitable way. In the sequel, we mainly talk about region 4 and note that the same reasoning also holds for region 2.

For a properly chosen δ, we intend to show that the size of R_4 is at most $2w$. To this end, we define the set $\bar{R}_4$ as

$$\bar{R}_4 = \{i \mid i + w \in R_4, i \in R_4\}, \tag{138}$$

and show that for a properly chosen δ the set $\bar{R}_4$ has size at most w. Clearly, this proves that the set R_4 has size at most $2w$. In order to show this, we first assume the contrary, i.e., the set $\bar{R}_4$ has size at least $w+1$, and then reach a contradiction. So assuming $\bar{R}_4$ has size at least $w+1$, let i_0 be the smallest position in $\bar{R}_4$. For $i \geq p$ we define y_i as

$$y_i = \ln(\frac{\ell_i}{2}). \tag{139}$$

We also define the two functions $h_1(x)$ and $h_2(x)$ as

$$h_1(x) = -\frac{K}{2^{K-1}}(1-x)^{K-1},$$
$$h_2(x) = \exp(x).$$

By using the conservation equations (86) we deduce that

$$y_i = \frac{1}{w}\sum_{k=0}^{w-1} \alpha_{i-k} h_1\big(\frac{1}{w}\sum_{d=0}^{w-1} h_2(y_{i-k+d})\big), \;\; \forall i \geq i_0. \tag{140}$$

Now, let us define the potential function

$$\bar{\Phi}(\{x_i\}) = \sum_{i=i_0}^{L+w-2} \big\{1 + x_i h_2(x_i) - H_2(x_i) - \alpha_i H_1(\frac{1}{w}\sum_{d=0}^{w-1} H_2(x_{i+d}))\big\}, \tag{141}$$

where the functions H_1, H_2 are given as

$$H_1(x) = \int h_1(x)dx = \frac{1}{2^{K-1}}(1-x)^K, \tag{142}$$

$$H_2(x) = \int h_2(x)dx = \exp(x). \tag{143}$$

It is easy to check that

$$(140) \Rightarrow \frac{\partial \bar{\Phi}}{\partial x_i}\Big|_y = 0, \;\; \forall i : i_0 + w \leq i \leq L + w - 2, \tag{144}$$

where by $\frac{\partial \bar{\Phi}}{\partial x_i}\Big|_y$ we mean the partial derivative of $\bar{\Phi}$ (with respect to x_i) computed at the profile $\{y_i\}_{i \geq i_0}$. The idea is now to use the relation (144) to get a contradiction with the fact that the set $\bar{R}_4$ has size at least $w+1$. In this regard, we define the shifted profile $\{Sy_i\}_{i\geq i_0}$ as

$$(Sy)_i = y_{i+1}, \forall i \geq i_0. \tag{145}$$

We first note that since $\{y_i\}_{i\geq i_0}$ is an increasing profile, then for $i \geq i_0$ we have that $(Sy)_i \geq y_i$. One can telescopically write (bearing in mind that $(Sy)_{L+w-2} = y_{L+w-1} = 0$)

$$\bar{\Phi}(\{Sy_i\}) - \bar{\Phi}(\{y_i\}) = -\Big(1 + y_{i_0} h_2(y_{i_0}) - H_2(y_{i_0}) - \alpha H_1(\frac{1}{w}\sum_{d=0}^{w-1} h_2(y_{i_0+d}))\Big) - \alpha H_1(\frac{1}{w}\sum_{d=0}^{w-1} h_2(y_{L+d}))$$

$$\leq -\Big(1 + y_{i_0} h_2(y_{i_0}) - H_2(y_{i_0}) - \alpha H_1(\frac{1}{w}\sum_{d=0}^{w-1} h_2(y_{i_0+d}))\Big). \tag{146}$$

Now, note that since $\bar{R}_4$ is assumed have size larger than w, then from the definition of $\bar{R}_4$ and i_0 it is easy to see that

$$\ell_i \in [\ell_u - \delta, \ell_u + \delta], \ \ \forall i \in \{i_0, \cdots, i_0 + w - 1\}. \tag{147}$$

By letting $y_u = \ln \frac{\ell_u}{2}$, we obtain for $i \in \{i_0, \cdots, i_0 + w - 1\}$

$$|h_2(y_i) - h_2(y_u)| = |H_2(y_i) - H_2(y_u)| = \frac{1}{2}|\ell_i - \ell_u| \leq \frac{\delta}{2}. \tag{148}$$

Also,

$$\begin{aligned}|y_i h_2(y_i) - y_u h_2(y_u)| &= \frac{1}{2}|\ell_i \ln \frac{\ell_i}{2} - \ell_u \ln \frac{\ell_u}{2}| \\ &\leq \delta \{ \sup_{x \in [\ell_u - \delta, \ell_u + \delta]} |1 + \ln \frac{x}{2}| \} \\ &\leq \delta \ln \frac{2}{\ell_u - \delta}. \end{aligned} \tag{149}$$

We further notice that

$$\sup_{x \in [0,1]} \big|H_1'(x)\big| = \frac{K}{2^{K-1}} \tag{150}$$

Now by (146), (148), (149) and (150) we conclude that

$$\bar{\Phi}(\{Sy_i\}) - \bar{\Phi}(\{y_i\}) \leq -\big(1 + y_u h_2(y_u) - H_2(y_u) + \alpha H_1(h_2(y_u))\big) + (\frac{1}{2} + \ln \frac{2}{\ell_u - \delta} + \frac{\alpha K}{2^K})\delta. \tag{151}$$

Now, as $\alpha < \alpha_{\text{cUCP}}$, from (92) we obtain that

$$y_u g(y_u) - G(y_u) + F(g(y_u)) \geq y_s g(y_s) - G(y_s) + F(g(y_s)) \triangleq \Delta(\alpha, K) > 0. \tag{152}$$

Thus by fixing δ to be

$$\delta = \delta(\alpha, K) \triangleq \min\{\exp(-\frac{\alpha K}{2^{K-1}}), \frac{\Delta}{4(\frac{1}{2} + \ln 2 + 3\frac{\alpha K}{2^K})}\}, \tag{153}$$

we first obtain from part (1) of the lemma that $\ln \frac{2}{\ell_u - \delta} \leq \ln 2 + \frac{\alpha K}{2^{K-1}}$, and also we conclude from (153), (152) and (151) that

$$\bar{\Phi}(\{S\bar{\ell}_i\}) - \bar{\Phi}(\{\bar{\ell}_i\}) \leq -\frac{3\Delta}{4}. \tag{154}$$

Note here that the choice of δ is merely dependent on α and K, and hence it is independent of the dynamics of the UC algorithm.

We now show that (154) cannot be true provided that the conservation equations (140) hold and w is large enough. The idea is compute the quantity $\big|\bar{\Phi}(\{Sy_i\}) - \bar{\Phi}(\{y_i\})\big|$ in a different was than above and show that its value can be made smaller than $\frac{\delta}{2}$ by increasing w, and hence we get a contradiction with (154). As we have assume that the size of $\bar{R}_4$ is larger than w, then $i_0 + w - 1 \in \bar{R}_4$. Consequently, we have

$$\ell_i \in [\ell_u - \delta, \ell_u + \delta], \forall i \in \{i_0, \cdots, i_{0+2w-1}\}.$$

Now, in a similar was as in (147)-(151) we can write

$$\Big|\big(1+y_{i_0+w-1}h_2(y_{i_0+w-1})-H_2(y_{i_0+w-1})-\alpha_{i_0+w-1}H_1(\frac{1}{w}\sum_{d=0}^{w-1}H_2(y_{i_0+w-1+d}))\big)$$
$$-\big(1+y_{i_0}h_2(y_{i_0})-H_2(y_{i_0})-\alpha_{i_0}H_1(\frac{1}{w}\sum_{d=0}^{w-1}H_2(y_{i_0+d}))\big)\Big|$$
$$\leq (\frac{1}{2}+\ln\frac{2}{\ell_u-\delta}+\frac{\alpha K}{2^K})\delta \overset{(153)}{\leq} \frac{\Delta}{4}.$$

Also, by using the mean-value theorem we know that there exists a profile $\{y^*{}_i\}_{i\geq i_0}$ such that for $i\geq i_0$ we have $y_i\leq y^*{}_i\leq y_{i+1}$ and

$$\big|\bar{\Phi}(\{Sy_i\})-\bar{\Phi}(\{y_i\})\big|\leq\frac{\Delta}{4}+\sum_{i\geq i_0+w}\frac{\partial\bar{\Phi}}{\partial x_i}\Big|_y(y_{i+1}-y_i)+\sum_{i,j\geq i_0+w}\frac{\partial^2\bar{\Phi}}{\partial x_i\partial x_j}\Big|_{y^*}(y_{i+1}-y_i)(y_{j+1}-y_j). \quad (155)$$

Here, by $\frac{\partial^2\bar{\Phi}}{\partial x_i\partial x_j}\Big|_{y^*}$ we mean the partial second derivative of $\bar{\Phi}$ given in (141) (with respect to x_i and x_j) computed at the profile $\{y^*{}_i\}_{i\geq i_0}$. Now, by using (144) we conclude that

$$\big|\bar{\Phi}(\{Sy_i\})-\bar{\Phi}(\{y_i\})\big|\leq\frac{\Delta}{4}+\sum_{i,j\geq i_0+w}\frac{\partial^2\bar{\Phi}}{\partial x_i\partial x_j}\Big|_{y^*}(y_{i+1}-y_i)(y_{j+1}-y_j). \quad (156)$$

We now bound the right-hand side of (156). For this purpose let us decompose the function $\bar{\Phi}$ into two parts

$$\bar{\Phi}(\{x_i\})=\underbrace{\sum_{i\geq i_0}1+x_ih_2(x_i)-H_2(x_i)}_{\bar{\Phi}_1}-\underbrace{\sum_{i\geq i_0}\alpha_iH_1(\frac{1}{w}\sum_{d=0}^{w-1}h_2(x_{i+d}))}_{\bar{\Phi}_2}. \quad (157)$$

Consequently,

$$\frac{\partial^2\bar{\Phi}}{\partial x_i\partial x_j}=\frac{\partial^2\bar{\Phi}_1}{\partial x_i\partial x_j}-\frac{\partial^2\bar{\Phi}_2}{\partial x_i\partial x_j}. \quad (158)$$

For the first part $\bar{\Phi}_1$, we have

$$\left|\frac{\partial^2(1+x_ih_2(x_i)-H_2(x_i))}{\partial x_i\partial x_j}\right|\leq\mathbb{1}\{i=j\}\sup_{x\in(-\infty,0]}h_2'(x)\leq\mathbb{1}\{i=j\}. \quad (159)$$

Hence, we conclude that

$$\Big|\sum_{i,j\geq i_0+w}\frac{\partial^2\bar{\Phi}_1}{\partial x_i\partial x_j}\Big|_{y^*}(y_{i+1}-y_i)(y_{j+1}-y_j)\Big|\leq\sum_{i\geq i_0}(y_{i+1}-y_i)(y_{i+1}-y_i)$$
$$\overset{(a)}{\leq}\frac{1}{w}\frac{\alpha K}{2^{K-1}}\sum_{i\geq i_0}(y_{i+1}-y_i)$$
$$\leq\frac{1}{w}\frac{\alpha K}{2^{K-1}}(-y_{i_0})$$

$$\overset{(b)}{\leq} \frac{1}{w}(\frac{\alpha K}{2^{K-1}})^2. \tag{160}$$

Here, step (a) follows from the fact that by the conservation equations in (140), we can easily see that for any $i \geq i_0$ we have

$$\begin{aligned} y_{i+1} - y_i &= \frac{1}{w}\sum_{k=0}^{w-1}\Big\{\alpha_{i+1-k}h_1\big(\frac{1}{w}\sum_{d=0}^{w-1}h_2(y_{i+1-k+d})\big) - \alpha_{i-k}h_1\big(\frac{1}{w}\sum_{d=0}^{w-1}h_2(y_{i-k+d})\big)\Big\} \\ &\leq -\frac{\alpha_{i-(w-1)}}{w}h_1\big(\frac{1}{w}\sum_{d=0}^{w-1}h_2(y_{i-(w-1)+d})\big) \\ &\leq \frac{\alpha}{w}\sup_{x\in[0,1]} -h_1(x) \\ &= \frac{\alpha K}{w 2^{K-1}}. \end{aligned}$$

Step (b) follows from part (1) of the lemma and (139).

For the second part $\bar{\Phi}_2$, one can write for $i, j, r \in \{i_0, \cdots, L+w-2\}$ (we assume here that $i \leq j$):

$$\begin{aligned} &\Big|\frac{\partial}{\partial x_j}\frac{\partial}{\partial x_i}H_1(\frac{1}{w}\sum_{d=0}^{w-1}h_2(x_{r+d}))\Big| \\ &= \Big|\mathbb{1}\{j-w \leq r \leq i\}\Big(\frac{h_2''(x_i)}{w}H_1'(\frac{1}{w}\sum_{d=0}^{w-1}h_1(x_{r+d}))\mathbb{1}\{i=j\} + \frac{h_2'(x_i)h_2'(x_j)}{w^2}H_1''(\frac{1}{w}\sum_{d=0}^{w-1}h_2(x_{r+d}))\Big)\Big| \\ &\leq \mathbb{1}\{j-w \leq r \leq i\}\Big(\frac{\theta^2}{w}\mathbb{1}\{i=j\} + \frac{\theta^3}{w^2}\Big), \end{aligned}$$

where

$$\theta = \max\{\sup_{x\in[0,1]}|H_1'(x)|, \sup_{x\in[0,1]}|H_1''(x)|, \sup_{x\in(-\infty,0]}|h_2'(x)|, \sup_{x\in(-\infty,0]}|h_2''(x)|\} = \max\{1, \frac{K(K-1)}{2^K}\}.$$

Therefore,

$$\begin{aligned} &\Big|\sum_{i,j\geq i_0+w}\frac{\partial^2\bar{\Phi}_2}{\partial x_i\partial x_j}\Big|_{y^*}(y_{i+1}-y_i)(y_{j+1}-y_j)\Big| \\ &\leq \alpha\sum_{i\geq i_0}\theta^2(y_{i+1}-y_i)^2 + \alpha\sum_{i\geq i_0}\sum_{j=i-w}^{i+w}\frac{\theta^3}{w}(y_{i+1}-y_i)(y_{j+1}-y_j) \\ &\leq \frac{\alpha\theta^2}{w}\frac{\alpha K}{2^{K-1}}\sum_{i\geq i_0}(y_{i+1}-y_i) + \frac{\alpha\theta^3}{w}\sum_{i\geq i_0}(y_{i+1}-y_i)(y_{i+w}-y_{i-w}) \\ &\leq \alpha\frac{\alpha\theta^2}{w}(\frac{\alpha K}{2^{K-1}})^2 + \frac{\alpha\theta^3}{w}(\frac{\alpha K}{2^{K-1}})^2 = \frac{1}{w}(\frac{\alpha K}{2^{K-1}})^2(\theta^2+\theta^3). \end{aligned} \tag{161}$$

Finally, as a consequence of (156), (158), (160) and (161) we get that

$$\big|\bar{\Phi}(\{S\bar{\ell}_i\}) - \bar{\Phi}(\{\bar{\ell}_i\})\big| \leq \frac{\varDelta}{4} + (\frac{\alpha K}{2^{K-1}})^2\frac{1+\alpha(\theta^2+\theta^3)}{w}. \tag{162}$$

Hence, by choosing

$$w > w_0(\alpha, K) \triangleq 4(\frac{\alpha K}{2^{K-1}})^2 \frac{1+\alpha(\theta^2+\theta^3)}{\Delta}, \tag{163}$$

we get that

$$\left|\bar{\Phi}(\{S\bar{\ell}_i\}) - \bar{\Phi}(\{\bar{\ell}_i\})\right| \leq \frac{\Delta}{2}. \tag{164}$$

The above inequality contradicts (154) when $\Delta > 0$ (or equivalently $\alpha < \alpha_{\mathrm{cUC}}$). Hence, the assumption that $\bar{R}_4$ has size larger than w contradicts the conservation equation (144) and part (5) is proved.

Proof of part (6): our objective here is to find a suitable candidate for the constants c_1 and c_2 such that the result of part (5) holds. Let us first prove part (6) for $\epsilon = \epsilon_0$ given as

$$\epsilon_0 = \min\{1, (\frac{2^{2K-3}}{\alpha K})^{\frac{1}{K-2}}\} \Longrightarrow \frac{\alpha K}{2^{2K-2}}(\epsilon_0)^{K-2} \leq \frac{1}{2}. \tag{165}$$

We first show that there exists a constant c_0 such that $I_{\epsilon_0} \leq c_0 w$. We consider two cases. In the first case, we assume that $\ell_u + \delta \geq 2 - \epsilon_0$. In this case, it is easy to see that $I_{\epsilon_0} - 1$ falls in one of the regions $R_1 - R_4$. Now, from parts (4)-(5) we have

$$|R_1| + |R_2| + |R_3| + |R_4| \leq \zeta w + 2w + \zeta w + 2w = (4 + 2\zeta)w. \tag{166}$$

Hence, by choosing

$$c_0 \geq (4 + 2\zeta) + 1, \tag{167}$$

we deduce the $I_{\epsilon_0} \leq c_0 w$ for the first case. In the second case, we assume that $\ell_u + \delta < 2 - \epsilon_0$. In this case I_{ϵ_0} falls inside the region R_5. Define the region of positions D as

$$D = \{i \geq p \mid \ell_u + \delta \leq \ell_i \leq 2 - \epsilon_0\}. \tag{168}$$

We further have

$$I_{\epsilon_0} \leq |R_1| + |R_2| + |R_3| + |R_4| + |D| \leq (4 + 2\zeta)w + |D|. \tag{169}$$

Here, we note that exactly as in the proof of part (4), there exists a constant $r = r(\alpha, K)$ such that

$$\mid D \mid \leq rw. \tag{170}$$

As a result, it is clear from (169) and (170) that we have for both of the above cases

$$I_{\epsilon_0} \leq (2\zeta + 5 + r)w. \tag{171}$$

Let us now prove part (6) for an arbitrary choice of ϵ. If $\epsilon > \epsilon_0$, then (171) is also a bound for I_ϵ because I_ϵ is clearly decreasing in ϵ. Assume now that $\epsilon < \epsilon_0$ and consider the region of positions E defined as

$$E = \{i \geq p \mid 2 - \epsilon_0 \leq \ell_i \leq 2 - \epsilon\}. \tag{172}$$

We have from (171)

$$I_\epsilon = I_{\epsilon_0} + \mid E \mid \leq (5 + 2\zeta + r)w + |E|. \tag{173}$$

It thus remains to find an upper bound on the size of E. Consider a position $i \in E$, we now note that for $m \in \mathbb{N}$

$$2 - \ell_{i+mw} \leq (2 - \ell_i)^{(K-1)^m} (\frac{\alpha K}{2^{2K-2}})^{(K-1)^{m-1}}.$$

This relation follows easily by m times using the inequality in part (2) of the lemma. Using this bound, we have

$$\begin{aligned}2-\ell_{i+mw} &\leq (2-\ell_i)^{(K-1)^m}(\frac{\alpha K}{2^{2K-2}})^{(K-1)^{m-1}} \\ &= (2-\ell_i)^{(K-1)^{m-1}}(\frac{\alpha K}{2^{2K-2}}(2-\ell_i))^{(K-1)^m-(K-1)^{m-1}} \\ &\overset{(a)}{\leq} (\frac{\alpha K}{2^{2K-2}}(2-\ell_i))^{(K-1)^m-(K-1)^{m-1}} \\ &\overset{(b)}{\leq} (\frac{1}{2})^{(K-1)^m-(K-1)^{m-1}} \\ &\overset{(c)}{\leq} (\frac{1}{2})^{2^{m-1}}. \end{aligned} \tag{174}$$

Here, step (a) follows from the fact that by using (165) and (172) we deduce that for $i \in E$ we have $\ell_i \geq 1$ and hence $2-\ell_i \leq 1$. Again by using (165) and (172) we deduce that for $i \in E$ we have $\frac{\alpha K}{2^{2K-2}}(2-\ell_i) \leq \frac{1}{2}$ and hence step (b) follows. Also, step (c) follows from the fact that $K \geq 3$. Now, for a position $i \in E$, it is apparent from (174) that if we choose

$$m = \lceil 1+\log_2(\log_2 \frac{1}{\epsilon})\rceil \Longrightarrow (\frac{1}{2})^{2^{m-1}} \leq \epsilon,$$

then we have $2-\ell_{i+mw} < \epsilon$ and hence $i+mw \notin E$. Therefore we have

$$|E| \leq \lceil w(1+\log_2(\log_2 \frac{1}{\epsilon}))\rceil,$$

and as a result we obtain from (173) that

$$\begin{aligned} I_\epsilon &\leq w(5+2\zeta+r+2+\log_2(\log_2 \frac{1}{\epsilon})) \\ &\leq w(\underbrace{7+2\zeta+r-\frac{\log(\log 2)}{\log 2}}_{\triangleq c_1(\alpha,K)} + \underbrace{\frac{1}{\log 2}}_{\triangleq c_2(\alpha,K)} \log(\log \frac{1}{\epsilon})). \end{aligned}$$

Proof of Lemma 4 Let us first explain what is the main approach behind the proof. Consider the vector $\bar{d} = \{d_i\}_{0\leq i\leq L+w-1}$ defined as follows. For $i \geq 0$,

$$d_i = w(\ell_{i+1}-\ell_i). \tag{175}$$

We first note that all the entries of the vector d are strictly positive. In what follows, we show that there exists a constant $\delta > 0$, such that the vector $A^n d$ decays to the all-zero vector faster than $(1-\delta)^n$. Using this, we conclude that the largest eigenvalue of the matrix A should be less than $(1-\delta)$.

We begin by defining for $n \in \mathbb{N}$, the *ratio* profile $\bar{\gamma}_n = \{\gamma_{n,i}\}_{0\leq i\leq L+w-2}$ as

$$\gamma_{n,i} = \begin{cases} 0 & i < p, \\ \frac{(A^n d)_i}{d_i} & i \geq p. \end{cases} \tag{176}$$

where by $(A^n d)_i$ we mean the i-th entry of the vector $A^n d$. Also for $n = 0$, we let

$$\gamma_{0,i} = \mathbb{1}_{\{i\geq p\}}. \tag{177}$$

The proof consists of four steps. For technical reasons that will be clear later on, we begin by confining our analysis to positions i that are inside a specific region defined as follows. Let $\epsilon > 0$ be a (small) constant, the value of which we will specify later in the proof. Define the integer T_ϵ as

$$T_\epsilon = \text{argmin}\{i \geq p \mid \forall j \geq i : \, \ell_{j+w} \geq 2 - \epsilon\}. \tag{178}$$

We first confine our analysis to the region $0 \leq i \leq T_\epsilon$. Note here that the smaller we make ϵ, the larger the value of T_ϵ will be. Also, in the last step of the proof, we show that by choosing a sufficiently small ϵ, all the final results are valid for the complete region $0 \leq i \leq L + w - 2$.

Step 1: Recursive bounds for the ratio profiles: In this step, we intend to bound the ratio profile $\bar{\gamma}_{n+1}$, in terms of the profile $\bar{\gamma}_n$.

We first write

$$\begin{aligned}(A^{n+1}d)_i &= (A(A^n d))_i \\ &= \sum_{j=0}^{L+w-2} A_{i,j}(A^n d)_j \\ &= \sum_{j=0}^{L+w-2} A_{i,j}\gamma_{n,j}d_j.\end{aligned}$$

Now, on one hand, by using (63) we have for $i \geq p$,

$$\begin{aligned}(A^{n+1}d)_i &= \ell_i \frac{\alpha K(K-1)}{2^K w} \sum_{j=i-w+1}^{i+w-1} \frac{\gamma_{n,j}d_j}{w} \sum_{k=|j-i|}^{w-1} \big(1 - \frac{1}{w}\sum_{d=0}^{w-1} \frac{\ell_{\max(i,j)-k+d}}{2}\big)^{K-2} \\ &= \ell_i \frac{\alpha K(K-1)}{2^K} \frac{1}{w} \sum_{k=0}^{w-1} \big(\frac{1}{w}\sum_{s=0}^{w-1} \gamma_{n,i-k+s}d_{i-k+s}\big)\big(1 - \frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}. \end{aligned} \tag{179}$$

On the other hand, from the conservation equations (86) we can write

$$\begin{aligned}\ln \frac{\ell_{i+1}}{\ell_i} =& \frac{\alpha K}{2^{K-1}} \frac{1}{w} \sum_{k=0}^{w-1} \big\{(1 - \frac{1}{w}\sum_{d=0}^{w-1} \frac{\ell_{i-k+d}}{2})^{K-1} - (1 - \frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i+1-k+d}}{2})^{K-1}\big\} \\ &+ \int_0^t 2\delta_{pi}\ell_i^{-1} dt, \end{aligned} \tag{180}$$

We now intend to bound the expression of (180). To simplify notation, let us first define

$$x_{i,k} = 1 - \frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}, \tag{181}$$

$$t_{i,k} = x_{i,k} - x_{i+1,k} = \frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i+1-k+d} - \ell_{i-k+d}}{2} = \frac{1}{w}\frac{\ell_{i+w-k} - \ell_{i-k}}{2} = \frac{1}{w^2}\sum_{d=0}^{w-1}\frac{d_{i-k+d}}{2}. \tag{182}$$

We first note that for $i \leq T_\epsilon$ we have

$$\frac{t_{i,k}}{x_{i,k}} = \frac{\sum_{d=0}^{w-1}\frac{\ell_{i+1-k+d}-\ell_{i-k+d}}{2}}{\sum_{d=0}^{w-1}\frac{2-\ell_{i-k+d}}{2}}$$

$$= \frac{\frac{\ell_{i-k+w} - \ell_{i-k}}{2}}{\sum_{d=0}^{w-1} \frac{2-\ell_{i-k+d}}{2}}$$
$$\leq \frac{2}{w\epsilon}, \tag{183}$$

where the last step follows from the definition of T_ϵ in (178) and the fact that $i \leq T_\epsilon$. Let us assume here that the value of w is chosen large enough so that

$$w \geq \frac{2^K}{\epsilon}. \tag{184}$$

Then, we clearly have $\frac{t_{i,k}}{x_{i,k}} < 1$ and after some simple manipulation we can write

$$\begin{aligned}
\frac{1}{w}\sum_{k=0}^{w-1} x_{i,k}^{K-1} - x_{i+1,k}^{K-1} &= \frac{1}{w}\sum_{k=0}^{w-1} x_{i,k}^{K-1} - (x_{i,k} - t_{i,k})^{K-1} \\
&= \frac{1}{w}\sum_{k=0}^{w-1}\sum_{s=1}^{K-1} (-1)^{s+1}\binom{K-1}{s} x_{i,k}^{K-1-s} t_{i,k}^{s} \\
&= \frac{1}{w}\sum_{k=0}^{w-1} x_{i,k}^{K-2} t_{i,k}\big\{(K-1) - \sum_{s=2}^{K-1}(-1)^s\binom{K-1}{s}(\frac{t_{i,k}}{x_{i,k}})^s\big\} \\
&\geq \frac{1}{w}\sum_{k=0}^{w-1} x_{i,k}^{K-2} t_{i,k}\big\{(K-1) - \frac{t_{i,k}}{x_i,k}\sum_{s=2}^{K-1}\binom{K-1}{s}\big\} \\
&\overset{(a)}{\geq} \frac{1}{w}\sum_{k=0}^{w-1} x_{i,k}^{K-2} t_{i,k}(K-1-\frac{2^K}{w\epsilon}) \\
&\geq \frac{K-1}{w}\sum_{k=0}^{w-1} x_{i,k}^{K-2} t_{i,k}(1-\frac{2^K}{w\epsilon})
\end{aligned} \tag{185}$$

where step (a) follows from (183). As a result of inequalities (180)-(185) we obtain for $i \leq T_\epsilon$

$$\ln\frac{\ell_{i+1}}{\ell_i} \geq \frac{\alpha K(K-1)}{2^K}(1-\frac{2^K}{w\epsilon})\frac{1}{w}\sum_{k=0}^{w-1}(\frac{1}{w^2}\sum_{d=0}^{w-1} d_{i-k+d})\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2} + \int_0^t 2\delta_{pi}\ell_i^{-1}dt. \tag{186}$$

Now, using the fact that $\frac{\ell_{i+1}}{\ell_i} = 1 + \frac{d_i}{w\ell_i}$ and the inequality $\ln(1+x) \leq x$, we have

$$\ln\frac{\ell_{i+1}}{\ell_i} \leq \frac{d_i}{w\ell_i}. \tag{187}$$

This inequality together with (186) yields

$$d_i \geq \frac{\alpha K(K-1)}{2^K}(1-\frac{2^K}{w\epsilon})\frac{1}{w}\sum_{k=0}^{w-1}(\frac{1}{w}\sum_{d=0}^{w-1} d_{i-k+d})\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2} \tag{188}$$

Finally, by using (179) and (188), we obtain for $0 \leq i \leq T_\epsilon$

$$\gamma_{n+1,i} = \frac{(A^{n+1}d)_i}{d_i} \leq \frac{1}{(1-\frac{2^K}{w\epsilon})} \frac{\sum_{k=0}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}\big(\sum_{d=0}^{w-1}\gamma_{n,i-k+d}d_{i-k+d}\big)}{\sum_{k=0}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}\big(\sum_{d=0}^{w-1}d_{i-k+d}\big)} \tag{189}$$

Step2: Upper bounds on $\bar{\gamma}_1$ and $\bar{\gamma}_2$: As we will see in the following, and this is also confirmed by numerical experiments, the behavior of the ration profile $\bar{\gamma}_1$ depends heavily on the value of ℓ_p. Consequently, we consider the following two cases which depend on the value of ℓ_p and provide an upper bound on $\bar{\gamma}_1$ for each case. We then use this upper bound together with the recursion (189) to provide an upper bound for $\bar{\gamma}_2$ (independent of the cases). We then use this upper bound in later steps of the proof.

Case1: We assume in this case that

$$\ell_{p+1} \leq 3\ell_p. \tag{190}$$

By using the recursion (189) and the fact that $\bar{\gamma}_0$ is the profile given in (177), we obtain after some simple manipulations that

$$\begin{aligned}
\gamma_{1,i} &\overset{(a)}{\leq} \frac{1}{(1-\frac{2^K}{w\epsilon})}\Bigg(1-\frac{\sum_{k=0}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}\big(\sum_{d=0}^{w-1}d_{i-k+d}\mathbb{1}_{\{i-k+d<0\}}\big)}{\sum_{k=0}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}\big(\sum_{d=0}^{w-1}d_{i-k+d}\big)}\Bigg)\\
&\overset{(b)}{=} \frac{1}{(1-\frac{2^K}{w\epsilon})}\Bigg(1-\frac{\sum_{k=i-p+1}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}w\ell_p}{\sum_{k=0}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}\big(\sum_{d=0}^{w-1}d_{i-k+d}\big)}\Bigg)\\
&\overset{(c)}{\leq} \frac{1}{(1-\frac{2^K}{w\epsilon})}(1-\frac{w-1-(i-p)}{2w}\ell_p)\\
&\overset{(d)}{\leq} \frac{1}{(1-\frac{2^K}{w\epsilon})}(1-\frac{w-1-(i-p)}{6w}\ell_{p+1})\\
&\overset{(e)}{\leq} \frac{1}{(1-\frac{2^K}{w\epsilon})}(1-\frac{w-1-(i-p)}{3w}\exp(-\frac{\alpha K}{2^{K-1}})).
\end{aligned} \tag{191}$$

Here, the relation (a) follows from (189) and (177). The relation (b) follows from the fact that $d_{p-1} = w\ell_p$ and $d_j = 0$ for $j < p-1$. The relation (c) follows from the inequality $\sum_{d=0}^{w-1} d_{i-k+d} \leq 2w$ and also from the fact that the sequence

$$y_k = (1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2})^{K-2}$$

is an increasing sequence in k, and hence,

$$\frac{\sum_{k=i-p+1}^{w-1} y_k}{\sum_{k=0}^{w-1} y_k} \geq \frac{w-1-(i-p)}{w}.$$

The relation (d) follows from (190) and finally the relation (e) follows from the first part of Lemma 3. As a result of the above series of inequalities, we deduce the following. For case 1, we have

$$\gamma_{1,i} \leq \frac{1}{(1-\frac{2^K}{w\epsilon})}\begin{cases}1-\frac{1}{12}\exp(-\frac{\alpha K}{2^{K-1}}) & p \leq i \leq p+\frac{w-1}{2},\\ 1 & p+\frac{w-1}{2} < i \leq T_\epsilon.\end{cases} \tag{192}$$

As we have for any i that $\gamma_{1,i} \leq \frac{1}{(1-\frac{2^K}{w\epsilon})}\gamma_{0,i}$, then a moment of thought reveals that with exactly the same argument as above we can also show that

$$\gamma_{2,i} \leq \frac{1}{(1-\frac{2^K}{w\epsilon})^2} \begin{cases} 1-\frac{1}{12}\exp(-\frac{\alpha K}{2^{K-1}}) & p \leq i \leq p+\frac{w-1}{2}, \\ 1 & p+\frac{w-1}{2} < i \leq T_\epsilon. \end{cases} \tag{193}$$

Case 2: Contrary to the first case, in this case we assume that

$$\ell_{p+1} > 3\ell_p. \tag{194}$$

As a result, we have that

$$\frac{\ell_{p+1}-\ell_p}{\ell_p} \geq 2.$$

Now, by using the fact that for $x \geq 2$ we have $\ln(1+x) \leq x - \frac{1}{2}$, we obtain that

$$\ln\frac{\ell_{p+1}}{\ell_p} = \ln(1+\frac{\ell_{p+1}-\ell_p}{\ell_p}) \leq \frac{\ell_{p+1}-\ell_p}{\ell_p} - \frac{1}{2},$$

and hence

$$\ln\frac{\ell_{p+1}}{\ell_p} \leq \frac{d_p}{w\ell_p} - \frac{1}{2}. \tag{195}$$

Now, by using (195) and (186) we obtain

$$d_p \geq \frac{w\ell_p}{2} + \ell_p\frac{\alpha K(K-1)}{2^K}(1-\frac{2^K}{w\epsilon})\frac{1}{w}\sum_{k=0}^{w-1}\Big(\sum_{d=0}^{w-1} d_{p-k+d}\Big)\Big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{p-k+d}}{2}\Big)^{K-2}. \tag{196}$$

Now, by using (196) and (179) we reach to the following bound on $\gamma_{1,p}$

$$\begin{aligned} \gamma_{1,p} &\leq \frac{1}{(1-\frac{2^K}{w\epsilon})}\frac{\sum_{k=0}^{w-1}\big(\sum_{d=0}^{w-1}\gamma_{0,p-k+d}d_{p-k+d}\big)\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{p-k+d}}{2}\big)^{K-2}}{\frac{w}{2}+\sum_{k=0}^{w-1}\big(\sum_{d=0}^{w-1}d_{p-k+d}\big)\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{ipk+d}}{2}\big)^{K-2}} \\ &\overset{(a)}{\leq} \frac{1}{(1-\frac{2^K}{w\epsilon})}\frac{2w}{\frac{w}{2}+2w} \\ &= \frac{4}{5(1-\frac{2^K}{w\epsilon})}, \end{aligned}$$

where (a) follows from the fact that $\sum_{d=0}^{w-1}\gamma_{0,p-k+d}d_{p-k+d} \leq \sum_{d=0}^{w-1} d_{p-k+d} \leq 2w$. Therefore, for the second case, the ratio profile $\bar{\gamma}_0$ is bounded from above as follows

$$\gamma_{1,i} \leq \frac{1}{(1-\frac{2^K}{w\epsilon})}\begin{cases}\frac{4}{5} & i=p, \\ 1 & p<i\leq T_\epsilon.\end{cases} \tag{197}$$

We now find an upper bound on the profile $\bar{\gamma}_2$ for this case. As the details of how we reach such an upper bound are exactly the same as in (191), we omit them here to avoid repeating long expressions. We have

$$\gamma_{2,i} \leq$$

$$
\begin{aligned}
&\frac{1}{(1-\frac{2^K}{w\epsilon})}\Big(\frac{\sum_{k=0}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}\big(\sum_{d=0}^{w-1}d_{i-k+d}\gamma_{1,i-k+d}\big)}{\sum_{k=0}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}\big(\sum_{d=0}^{w-1}d_{i-k+d}\big)}\Big)\\
&\overset{(a)}{\le}\frac{1}{(1-\frac{2^K}{w\epsilon})^2}\Big(1-\frac{\sum_{k=0}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}\sum_{d=0}^{w-1}\frac{1}{5}d_p\mathbb{1}_{\{i-k+d=p\}}}{\sum_{k=0}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}\big(\sum_{d=0}^{w-1}d_{i-k+d}\big)}\Big)\\
&\overset{(b)}{\le}\frac{1}{(1-\frac{2^K}{w\epsilon})^2}\Big(1-\frac{\sum_{k=i-p}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}\frac{2}{15}w\ell_{p+1}}{\sum_{k=0}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}\big(\sum_{d=0}^{w-1}d_{i-k+d}\big)}\Big)\\
&\le\frac{1}{(1-\frac{2^K}{w\epsilon})^2}(1-\frac{w-(i-p)}{w}\frac{4}{15}\ell_{p+1})\\
&\le\frac{1}{(1-\frac{2^K}{w\epsilon})^2}(1-\frac{w-(i-p)}{w}\frac{4}{15}\exp(-\frac{\alpha K}{2^{K-1}})).
\end{aligned}
\tag{198}
$$

Here, step (a) follows from (197). Step (b) follows from (194). The other steps follow exactly similar to the proof of (191) and hence we omit further explanations about them. As a result of the above set of inequalities, we have

$$
\gamma_{2,i}\le\frac{1}{(1-\frac{2^K}{w\epsilon})^2}\begin{cases}1-\frac{2}{15}\exp(-\frac{\alpha K}{2^{K-1}}) & p\le i\le p+\frac{w-1}{2},\\ 1 & p+\frac{w-1}{2}<i\le T_\epsilon.\end{cases}\tag{199}
$$

Let us now finalize the result of Step 2 of the proof. By using (193) and (199) we deduce that the ratio profile $\bar{\gamma}_2$ is bounded above as follows:

$$
\gamma_{2,i}\le\frac{1}{(1-\frac{2^K}{w\epsilon})^2}\begin{cases}1-\frac{1}{12}\exp(-\frac{\alpha K}{2^{K-1}}) & p\le i\le p+\frac{w-1}{2},\\ 1 & p+\frac{w-1}{2}<i\le T_\epsilon.\end{cases}\tag{200}
$$

Step3: A recursive bound on $\bar{\gamma}_n$: In this step we prove the following bound on $\bar{\gamma}_n$: for $n\ge 2$, we have

$$
\gamma_{n,i}\le\frac{1}{(1-\frac{2^K}{w\epsilon})^n}\begin{cases}1-c_3c_4^{n-2} & p\le i\le p+(n-1)\frac{w-1}{4},\\ \\ 1 & p+(n-1)\frac{w-1}{4}<i\le T_\epsilon,\end{cases}\tag{201}
$$

where the constants c_3 and c_4, that depend on α and K, are given as

$$
c_3=\frac{1}{12}\exp(-\frac{\alpha K}{2^{K-1}}),\tag{202}
$$

$$
c_4=\frac{1}{8c},\tag{203}
$$

and also the constant c is given in Lemma 5. We prove this statement by induction on n. For $n=2$ the result is clear due to (200). Let us assume (201) holds for $n=m$. We now show that it also holds for $n=m+1$. Consider a position $p\le i\le p+m\frac{w-1}{4}$. We can write

$$
\begin{aligned}
&(1-\frac{2^K}{w\epsilon})^{m+1}\gamma_{m+1,i}\\
&\overset{(a)}{\le}\frac{\sum_{k=0}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}\sum_{d=0}^{w-1}d_{i-k+d}\gamma_{m,i-k+d}(1-\frac{2^K}{w\epsilon})^m}{\sum_{k=0}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}\big(\sum_{d=0}^{w-1}d_{i-k+d}\big)}\\
&\overset{(b)}{\le}1-\frac{\sum_{k=0}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}\sum_{d=0}^{w-1}d_{i-k+d}c_3c_4^{m-2}\mathbb{1}_{\{i-k+d\le p+(m-1)\frac{w-1}{4}\}}}{\sum_{k=0}^{w-1}\big(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\big)^{K-2}\big(\sum_{d=0}^{w-1}d_{i-k+d}\big)}
\end{aligned}
$$

$$
\begin{aligned}
&\overset{(c)}{\le} 1 - \frac{\sum_{k=\frac{w-1}{2}}^{w-1}\left(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\right)^{K-2}\sum_{d=0}^{\frac{w-1}{4}} d_{i-k+d}c_3c_4^{m-2}}{\sum_{k=0}^{w-1}\left(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\right)^{K-2}\left(\sum_{d=0}^{w-1} d_{i-k+d}\right)}\\
&\overset{(d)}{\le} 1 - c_3c_4^{m-2}\frac{1}{4c}\frac{\sum_{k=\frac{w-1}{2}}^{w-1}\left(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\right)^{K-2}}{\sum_{k=0}^{w-1}\left(1-\frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\right)^{K-2}}\\
&\le 1 - c_3c_4^{m-2}\frac{1}{8c} = 1 - c_3c_4^{m-1}.
\end{aligned}
$$

Here, step (a) follows from (189). Step (b) follows from the induction hypothesis and (201). Step (c) follows from the fact that $i \le p + m\frac{w-1}{4}$. Step (d) follows by noticing from Lemma 5 that for integers $k_1 > k_2$ we have

$$
\sum_{d=0}^{w-1} d_{i-k_1+d} \le 4c \sum_{d=0}^{w-1} d_{i-k_1+d}.
$$

Finally, step (e) follows from the fact that the sequence

$$
y_k = (1 - \frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2})^{K-2}
$$

is an increasing sequence in k.

Step 4: Putting things together: Finally in this step we will complete the proof of the lemma. The main tools that we use here are the bound (201) and the following facts deduced from the the Perron-Frobenius formalism [39]. Consider an $r \times r$ matrix $X = [X_{i,j}]_{1\le i,j\le r}$ with non-negative entries.

(F1) There exist a number $\lambda_X \ge 0$ such that λ_X is itself an eigenvalue of X and any other eigenvalue λ of X (possibly complex) is smaller than λ_X in absolute value, $|\lambda| \le \lambda_X$. We call λ_X the *largest eigenvalue* of X.

(F2) We have

$$
\lambda_X \le \max_{1\le i\le r} \sum_{j=1}^{r} X_{i,j}. \tag{204}
$$

(F3) In addition, if X is symmetric, the value of λ_X can be computed as follows

$$
\lambda_X = \max_{\{y=(y_1,\cdots,y_r)\ \text{s.t.}\ \forall j:\ y_j\ge 0\}} \frac{y^TXy}{y^Ty}. \tag{205}
$$

(F4) If there exist integers $m \in \mathbb{N}$ such that all the entries of X^m are strictly positive, then λ_X is a simple eigenvalue of X. If such an assumption holds, we let v_X denote the eigenvector corresponding to λ_X.

(F5) With the assumptions of (F4), let d be a vector of size r such that $d_j > 0$ for $1 \le j \le r$. Then, there exists a constant $e > 0$ such that for $n \in \mathbb{N}$ we have

$$
X^n d = e\lambda_X^n v_X + o(r\lambda_X^n). \tag{206}
$$

We now proceed with the proof. Let B be a $(L + w - 1) \times (L + w - 1)$ matrix whose entries are given as follows.

$$
B_{i,j} = \begin{cases} A_{i,j} & 0 \le i,j \le T_\epsilon, \\ 0 & o.w. \end{cases} \tag{207}
$$

Also, let D be a $(L+w-1)\times(L+w-1)$ matrix whose entries are given as follows.

$$D_{i,j} = \begin{cases} A_{i,j} \; i,j \geq T_\epsilon - (w-1), \\ 0 \qquad\qquad\qquad\qquad o.w. \end{cases} \tag{208}$$

A moment of thought shows that B and D have non-negative entries and for $0 \leq i,j \leq L+w-1$ we have

$$A_{i,j} \leq B_{i,j} + D_{i,j}. \tag{209}$$

Let us now denote the largest eigenvalue of A, B and D by λ_A, λ_B and λ_D, respectively. Let us now show that

$$\lambda_A \leq \lambda_B + \lambda_D. \tag{210}$$

Form (37), it is easy to see that $A = LS$, where L is a diagonal matrix and S is a symmetric matrix. This is also true for the matrices B and D with the same diagonal matrix L, i.e., $B = LS_1$ and $D = LS_2$, where S_1 and S_2 are symmetric. Further, a moment of thought reveals that (i) the matrix $L^{-\frac{1}{2}}AL^{\frac{1}{2}} = L^{\frac{1}{2}}SL^{\frac{1}{2}}$ is a symmetric matrix and (ii) the matrices A and $L^{-\frac{1}{2}}AL^{\frac{1}{2}}$ have the same set of eigenvalues. Consequently, from the fact (F3) we obtain

$$\lambda_A = \max_{\{y=(y_1,\cdots,y_r) \text{ s.t. } \forall j:\, y_j \geq 0\}} \frac{y^T L^{-\frac{1}{2}} A L^{\frac{1}{2}} y}{y^T y}.$$

By repeating the exact same argument for B and D, we obtain that

$$\lambda_B = \max_{\{y=(y_1,\cdots,y_r) \text{ s.t. } \forall j:\, y_j \geq 0\}} \frac{y^T L^{-\frac{1}{2}} B L^{\frac{1}{2}} y}{y^T y},$$
$$\lambda_D = \max_{\{y=(y_1,\cdots,y_r) \text{ s.t. } \forall j:\, y_j \geq 0\}} \frac{y^T L^{-\frac{1}{2}} D L^{\frac{1}{2}} y}{y^T y}.$$

Now, by using the above relations for $\lambda_A, \lambda_B, \lambda_D$ and also the relation (209), the relation (210) follows easily.

The idea is now to show that if the value of ϵ is chosen suitably in terms of α and K, then there exists a constant $\delta = \delta(\alpha, K)$ such that

$$\lambda_B + \lambda_D \leq 1 - \delta, \tag{211}$$

and as a result, the whole proof is complete by noting the relation (210). Therefore, what remains to be done is to provide suitable upper bounds of λ_B and λ_D. We start with the matrix D. Note here that the value of ϵ is at our hands to choose. In other words, in order to prove that $\lambda_B + \lambda_D$ is strictly below 1 by a constant gap, we should "choose" a suitable value of ϵ. Hence, in the following we will gradually give several upper bounds (that only depend on α and K) on the value of ϵ. We then choose in the end a value of ϵ that satisfies all these bounds and show that for this value of ϵ the quantity $\lambda_B + \lambda_D$ is strictly below 1 by a gap that is only dependent of α and K.

To find an upper bound on λ_D, the idea here is to find an upper bound on the sum of the components of each row of D and then use the fact (F2). In this regard, we need to find suitable bounds on the entries of D. Consider a position $j \geq T_\epsilon$. From (178) we deduce that

$$\ell_{j+w} > 2 - \epsilon.$$

Assume here that we choose the value of ϵ to be

$$\epsilon < 1. \tag{212}$$

As a result,

$$\ln \frac{\ell_{j+w}}{2} > \ln(1 - \frac{\epsilon}{2}) > -\epsilon.$$

Now, from this inequality and part (3) of Lemma 3 we have

$$\frac{-\alpha}{2^{2K-1}}(\frac{2-\ell_{j+\frac{w}{2}}}{2})^{K-1} > -\epsilon \Longrightarrow 2 - \ell_{j+\frac{w}{2}} < (\epsilon\frac{2^{2K-1}}{\alpha})^{\frac{1}{K-1}}.$$

Now, again by choosing

$$(\epsilon\frac{2^{2K-1}}{\alpha})^{\frac{1}{K-1}} \leq 1 \Longrightarrow \epsilon \leq \frac{\alpha}{2^{2K-1}}, \tag{213}$$

we obtain that

$$\ln \frac{\ell_{j+\frac{w}{2}}}{2} > \ln(1 - \frac{(\epsilon\frac{2^{2K-1}}{\alpha})^{\frac{1}{K-1}}}{2}) > -(\epsilon\frac{2^{2K-1}}{\alpha})^{\frac{1}{K-1}},$$

and by using this inequality and part (3) of Lemma 3 we deduce that for $j \geq T_\epsilon$ we have

$$2 - \ell_j \leq \epsilon^{\frac{1}{(K-1)^2}}(\frac{2^{2K-1}}{\alpha})^{\frac{K}{(K-1)^2}}. \tag{214}$$

One can repeat the above argument once more to get a similar upper bound, as in the form of (214), for $2 - \ell_{j-w}$ and then for $2 - \ell_{j-2w}$ and so on. Since, the arguments are exactly as above, we do not repeat the details and only mention the net result: by choosing

$$\epsilon \leq (\frac{\alpha}{2^{2K-1}})^{K+K(K-1)^2+(K-1)^4} \triangleq c_5(\alpha, K). \tag{215}$$

we have for $j \geq T_\epsilon$

$$2 - \ell_{j-2w} \leq \epsilon^{\frac{1}{(K-1)^6}} \underbrace{(\frac{2^{2K-1}}{\alpha})^{\frac{K}{(K-1)^6}+\frac{K}{(K-1)^4}+\frac{K}{(K-1)^2}}}_{\triangleq c_6(\alpha,K)} = \epsilon^{\frac{1}{(K-1)^6}} c_6. \tag{216}$$

Equivalently, we can say that for $i \geq T_\epsilon - 2w$ we have

$$2 - \ell_i \leq \epsilon^{\frac{1}{(K-1)^6}} c_6. \tag{217}$$

Now, recall the definition of the matrix D from (208). For $i < T_\epsilon - w$, all the entries in the i-th row of D are equal to 0. For $i \geq T_\epsilon - w$, we first note from (63) that the sum of the entries of the i-th row is

$$\begin{aligned} \sum_{j=0}^{L+w-1} D_{i,j} &= \sum_{j=i-w+1}^{i-w+1} \frac{\alpha}{2^K}\frac{K(K-1)}{w}\ell_i \frac{1}{w}\sum_{k=|j-i|}^{w-1}\left(1 - \frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{\max(i,j)-k+d}}{2}\right)^{K-2} \\ &\leq \frac{\alpha}{2^K}\frac{K(K-1)}{w}\ell_i \sum_{k=0}^{w-1}\left(1 - \frac{1}{w}\sum_{d=0}^{w-1}\frac{\ell_{i-k+d}}{2}\right)^{K-2} \end{aligned}$$

$$\leq \frac{\alpha K(K-1)}{2^K}\ell_i(1-\frac{\ell_{i-w}}{2})^{K-2},$$

where the last step follows from the fact that the profile of literals $\{\ell_j\}_{j\geq 0}$ is non-decreasing in j. As a result, by using (217), provided that (215) holds, we obtain for $i \geq T_\epsilon - w$

$$\sum_{j=0}^{L+w-1} D_{i,j} \leq \epsilon^{\frac{K-2}{(K-1)^6}} \underbrace{(\frac{c_6}{2})^{K-2}\frac{K(K-1)}{2^{K-1}}}_{\triangleq c_7(\alpha,K)}. \tag{218}$$

Now, from the fact that for $i < T_\epsilon - w$ the i-th row of D has all its entries equal to 0, we deduce that

$$\max_{0\leq i\leq L+w-1} \sum_{j=0}^{L+w-1} D_{i,j} \leq \epsilon^{\frac{K-2}{(K-1)^6}} c_7. \tag{219}$$

Finally, by using the fact (F2) we conclude the following. Provided that (215) holds, we have

$$\lambda_D \leq \epsilon^{\frac{K-2}{(K-1)^6}} c_7. \tag{220}$$

We proceed by finding an upper bound on λ_B. Consider the vector $\bar{d}$ defined in (175). Note here for $i \geq p$ we have $d_i > 0$. The idea here is first to show that the vector $B^n d$ converges exponentially to 0 in n. From (207) we have for $n \in \mathbb{N}$

$$B^n d \leq A^n d, \tag{221}$$

and as a result, we obtain from (176) that for $i \geq 0$

$$(B^n d)_i \leq \gamma_{n,i} d_i. \tag{222}$$

From part (5) of Lemma 3 we can easily deduce that

$$T_\epsilon - p \leq w(c_1 + c_2 \log(\log \frac{1}{\epsilon})). \tag{223}$$

Consider now the integer m defined as

$$m_\epsilon = \lceil 8(c_1 + c_2 \log(\log \frac{1}{\epsilon}))\rceil. \tag{224}$$

It is easy to see from (223) that

$$m_\epsilon \geq \frac{4(T_\epsilon - p)}{w-1}. \tag{225}$$

Now, from the relations (222), (201) and the fact that $B_{i,j} = 0$ for $i, j > T_\epsilon$, we deduce that

$$B^{m_\epsilon} d \leq \frac{1 - c_3 c_4^{m_\epsilon - 2}}{(1-\frac{2^K}{w\epsilon})^{m_\epsilon}} d. \tag{226}$$

Let us proceed with further simplification of the above bound. Assume now that w is chosen to be

$$\frac{2^K}{w\epsilon} \leq \frac{1}{2} \Longrightarrow w \geq \frac{2^{K+1}}{\epsilon}. \tag{227}$$

By using the relation $\frac{1}{1-x} \leq 1 + 2x$ (for $x \leq \frac{1}{2}$), we obtain

$$\frac{1}{(1-\frac{2^K}{w\epsilon})^{m_\epsilon}} \leq (1+\frac{2^{K+1}}{w\epsilon})^{m_\epsilon} \leq 1 + 2^{m_\epsilon}\frac{2^{K+1}}{w\epsilon}.$$

As a result, we obtain from (226)

$$B^{m_\epsilon} d \leq (1+\frac{2^{m_\epsilon+K+1}}{w\epsilon})(1-c_3 c_4^{m\epsilon-2}). \tag{228}$$

Now, by choosing

$$\frac{2^{m_\epsilon+K+1}}{w\epsilon} \leq c_3 c_4^{m\epsilon-2} \Longrightarrow w \geq \frac{2^{m_\epsilon+K+1}}{\epsilon c_3 c_4^{m\epsilon-2}}, \tag{229}$$

we obtain from (228)

$$B^{m_\epsilon} d \leq 1 - (c_3 c_4^{m_\epsilon-2})^2 d. \tag{230}$$

Consequently, for any integer $n = m_\epsilon u$ we have

$$\begin{aligned} B^n d &= (B^{m_\epsilon})^u d \\ &\leq (1-(c_3 c_4^{m_\epsilon-2})^2)^u d \\ &= \left((1-(c_3 c_4^{m_\epsilon-2})^2)^{\frac{1}{m_\epsilon}}\right)^n d. \end{aligned} \tag{231}$$

We are now ready to use the result of the fats (F4) and (F5). But before that, the assumptions of the fact (F4) should be checked. Let $\tilde{B}$ be a $(T_\epsilon - p) \times (T_\epsilon - p)$ matrix defined as follows. For $i, j \leq T_\epsilon - p$

$$\tilde{B}_{i,j} = B_{i+p,j+p} \overset{(207)}{=} A_{i+p,j+p}. \tag{232}$$

It is easy to see from the definition of B in (207) that

$$B = \left[\begin{array}{c|c|c} \mathbf{0}_{p\times p} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \tilde{B} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0}_{(L-w-T_\epsilon)\times(L-w-T_\epsilon)} \end{array}\right]. \tag{233}$$

Here, by $\mathbf{0}$ we mean a matrix whose entries are all zero. It is also easy to check that for the integer $R = \lceil \frac{T_\epsilon - p}{w} \rceil$, all the entries of the matrix $\tilde{B}^R$ are strictly positive. Hence, the results of the facts (F4) and (F5) apply to the matrix $\tilde{B}$. It is now easy to see from (233) that the result of the facts apply also to the matrix B. That is, from fact (F4) we know that λ_B is a simple, and from the fact (F5) we know that for the vector d and $n \in \mathbb{N}$ we have

$$B^n d = c_8 \lambda_B^n v + o((T_\epsilon - p)\lambda_X^n), \tag{234}$$

where c_8 is a positive constant and v is the unique right eigenvector of B corresponding to λ_B. Now, from (231) and (234) we can easily see that

$$\lambda_B \leq (1-(c_3 c_4^{m_\epsilon-2})^2)^{\frac{1}{m_\epsilon}}. \tag{235}$$

Now, by using (220) and (235), we have

$$\lambda_B + \lambda_D \leq (1-(c_3 c_4^{m_\epsilon-2})^2)^{\frac{1}{m_\epsilon}} + \epsilon^{\frac{K-2}{(K-1)^6}} c_7$$

$$\leq 1-\frac{1}{m_\epsilon}(c_3c_4^{m_\epsilon-2})^2+\epsilon^{\frac{K-2}{(K-1)^6}}c_7, \tag{236}$$

where in the last step we have used the fact that for numbers $x,y\in[0,1]$, we have $(1-x)^y\leq 1-xy$. We proceed by showing that there exists a constant $c_9\triangleq c_9(\alpha,K)$ such that by choosing

$$\epsilon\leq c_9(\alpha,K), \tag{237}$$

we have

$$\epsilon^{\frac{K-2}{(K-1)^6}}c_7\leq\frac{1}{2}\Big(\frac{1}{m_\epsilon}(c_3c_4^{m_\epsilon-2})^2\Big). \tag{238}$$

To find such a candidate for c_9, we note that in order for (238) to hold, we must have

$$\frac{K-2}{(K-1)^6}\log\epsilon+\log c_7\leq-\log 2m_\epsilon+2\log c_3+2(m_\epsilon-2)\log c_4.$$

By rearranging the terms we get to

$$\log\frac{1}{\epsilon}\geq\frac{(K-1)^6}{K-2}\Big(\log\frac{2c_7c_4^4}{c_3^2}-2m_\epsilon\log c_4+\log m_\epsilon\Big),$$

and by using (224), we deduce that in order for (238) to hold, it is sufficient to have

$$\log\frac{1}{\epsilon}\geq\frac{(K-1)^6}{K-2}\Big(\log\frac{16c_7c_4^4}{c_3^2}-16(1+c_1)\log c_4-c_2\log c_4\log(\log\frac{1}{\epsilon})+\log(c_1+c_2\log(\log\frac{1}{\epsilon}))\Big). \tag{239}$$

Now, note here that all the constants c_1-c_8 defined above are positive constants which only depend on α and K. Also, it is easy to that if ϵ is sufficiently small, the relation (239) holds true. This proves the existence of a constant $c_9=c_9(\alpha,K)$ such that if choose ϵ according to (237), then the relation (239), and hence the relation (238), hold true. Now, from (215), (237), let us choose the value of ϵ to be

$$\epsilon=\min(c_5,c_9)\triangleq c_{10}(\alpha,K). \tag{240}$$

For this value of ϵ, by plugging (238) into (236) we obtain

$$\lambda_B+\lambda_D\leq 1-\underbrace{\frac{1}{16(c_1+c_2\log(\log(\frac{1}{c_{10}})))}(c_3c_4^{8(c_1+c_2\log(\log(\frac{1}{c_{10}})))-2})^2}_{\triangleq\delta(\alpha,K)}=1-\delta.$$

Of course, the above relation holds true provided that from (227), (229) and the value of $w_0(\alpha,K)$ given in Lemma 3, we have

$$w\geq\max\Big(\frac{2^{K+1}}{c_{10}},\frac{2^{K+1+8(c_1+c_2\log(\log(\frac{1}{c_{10}})))}}{c_{10}c_1c_2^{8(c_1+c_2\log(\log\frac{1}{c_{10}}))-2}},w_0(\alpha,K)\Big)\triangleq w_1(\alpha,K).$$

An Auxiliary Lemma: We now state and prove the Lemma used in steps 3 and 4 (in particular the value c in (203) is given in the following Lemma).

Lemma 5. *Define*

$$d_i \triangleq w(\ell_{i+1} - \ell_i).$$

Then, there exist a constant $c = c(\alpha, K)$, which only depends on α and K, such that the following holds

$$\max_{p \le i \le j} \frac{d_j}{d_i} \le c. \tag{241}$$

Proof. We first note that using the conservation equations (86) we have

$$\ln \frac{\ell_{i+1}}{\ell_i} = Q_i - Q_{i+1} + \int_0^t 2\delta_{pi} \ell_i^{-1} dt,$$

and by using (68), we obtain

$$\begin{aligned}\ln \frac{\ell_{i+1}}{\ell_i} =& \frac{\alpha K}{2^{K-1}} \big((1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\ell_{i+1-w+d}}{2})^{K-1} - (1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\ell_{i+1+d}}{2})^{K-1} \big) \\ &+ \int_0^t 2\delta_{pi} \ell_i^{-1} dt, \end{aligned} \tag{242}$$

The idea of the proof is to consider two different regions for the positions $i \ge p$ and provide a suitable candidate for the value c based on a careful analysis on these regions. Let us begin by defining i_0 to be

$$i_0 = \text{argmin}\big\{ i \ge p \mid \ell_{i+1} > \max\{2 - 4(\frac{2}{\alpha K})^{\frac{1}{K-2}}, 1\} \big\}. \tag{243}$$

Note here that by using the second part of Lemma 3 we deduce that

$$\forall i \ge i_0 : 2 - \ell_{i+w} < \frac{1}{2}(2 - \ell_i). \tag{244}$$

In order to find a suitable candidate for c in (241), we consider two cases for the positions i, j.

Case 1: We consider positions i, j such that $j \ge i \ge i_0 + w$. For such positions, on one hand we can write

$$\begin{aligned}\ln \frac{\ell_{i+1}}{\ell_i} &= \frac{\alpha K}{2^{K-1}} \big((\frac{1}{w} \sum_{d=0}^{w-1} \frac{2 - \ell_{i+1+d-w}}{2})^{K-1} - (\frac{1}{w} \sum_{d=0}^{w-1} \frac{2 - \ell_{i+1+d}}{2})^{K-1} \big) \\ &\overset{(244)}{\ge} \frac{\alpha K}{2^{K-1}} \big((\frac{1}{w} \sum_{d=0}^{w-1} \frac{2 - \ell_{i+1+d-w}}{2})^{K-1} - (\frac{1}{w} \sum_{d=0}^{w-1} \frac{2 - \ell_{i+1+d-w}}{4})^{K-1} \big) \\ &= \frac{\alpha K}{2^{K-1}} (1 - \frac{1}{2^{K-1}}) (\frac{1}{w} \sum_{d=0}^{w-1} \frac{2 - \ell_{i+1+d-w}}{2})^{K-1}. \end{aligned}$$

As a result, by noticing that $\frac{\ell_{i+1}}{\ell_i} = 1 + \frac{d_i}{w\ell_i}$ and using the first part of Lemma 3, we obtain

$$d_i \ge w\ell_i \frac{\alpha K}{2^{K-1}} (1 - \frac{1}{2^{K-1}}) (1 - \frac{1}{w} \sum_{d=0}^{w-1} \frac{\ell_{i+1+d-w}}{2})^{K-1}. \tag{245}$$

On the other hand, by using (242) for any position j, we have

$$\ln \frac{\ell_{j+1}}{\ell_j} \leq \frac{\alpha K}{2^{K-1}}(1 - \frac{1}{w}\sum_{d=0}^{w-1} \frac{\ell_{j+1-w+d}}{2})^{K-1}.$$

Now, as $j \geq i > i_0$, then by definition we have $\ell_j \geq 1$, hence $\ell_{j+1} - \ell_j \leq 1$ and hence $\frac{\ell_{j+1}-\ell_j}{\ell_j} \leq 1$. We thus have $\ln(1 + \frac{\ell_{j+1}-\ell_j}{\ell_j}) \geq \frac{1}{2}\frac{\ell_{j+1}-\ell_j}{\ell_j}$. As a result,

$$d_j \leq w\ell_j \frac{2\alpha K}{2^{K-1}}(1 - \frac{1}{w}\sum_{d=0}^{w-1} \frac{\ell_{j+1-w+d}}{2})^{K-1}. \tag{246}$$

Finally, for $j \geq i \geq i_0 + w$, we obtain from (245) and (246)

$$\frac{d_j}{d_i} \leq \frac{\ell_j}{\ell_i}(\frac{2^{K-1}}{2^{K-1}-1})\frac{(1 - \frac{1}{w}\sum_{d=0}^{w-1} \frac{\ell_{j+1-w+d}}{2})^{K-1}}{(1 - \frac{1}{w}\sum_{d=0}^{w-1} \frac{\ell_{i+1-w+d}}{2})^{K-1}},$$

which easily results in the following simple inequality for $j \geq i \geq i_0 + w$

$$\frac{d_j}{d_i} \leq 4. \tag{247}$$

<u>Case 2:</u> In this case, we consider positions i, j such that $j \geq i$ and $i < i_0 + w$. We first note that by using the relation (243) and part (3) of Lemma 3, we get that

$$\ell_{i_0+w} \leq 2\exp(-\alpha K 2^{-2K}\exp(-\alpha K 2^{-3K})). \tag{248}$$

Now, using the results of Lemma 3 and (248), we deduce that there exists a value $\zeta \triangleq \zeta(\alpha, K) > 0$ such that for $p \leq i \leq i_0 + w$ we have

$$\ell_{i+w} - \ell_i \geq \zeta. \tag{249}$$

Furthermore, by using (242) and the fact that for $x \geq y$ we have $x^{K-1} - y^{K-1} \geq (x-y)^{K-1}$, we can write for $i \geq p$

$$\begin{aligned}\ln \frac{\ell_{i+1}}{\ell_i} &\geq \frac{\alpha K}{2^{K-1}}(\frac{1}{w}\sum_{d=0}^{w-1} \frac{\ell_{i+1+d} - \ell_{i+1+d-w}}{2})^{K-1} \\ &\geq \frac{\alpha K}{2^{K-1}}\zeta^{K-1},\end{aligned}$$

and as a result, by using the first part of Lemma 3 we get for $p \leq i \leq i_0 + w$

$$d_i \geq w\frac{\alpha K}{2^{K-2}}\zeta^{K-1}\exp(-\frac{\alpha K}{2^{K-1}}) \triangleq D(\alpha, K), \tag{250}$$

and by noting that $d_j = \ell_{j+1} - \ell_j \leq 2$, we have for positions i, j such that $j \geq i$ and $i < i_0 + w$

$$\frac{d_j}{d_i} \leq \frac{2}{D(\alpha, K)}. \tag{251}$$

Finally, a candidate for the value of c in the lemma is $c = \max\{4, \frac{2}{D(\alpha,K)}\}$.