Ann. Inst. H. Poincaré D
Comb. Phys. Interact. 9 (2022), 435–471
DOI 10.4171/AIHPD/122

Moser–Tardos resampling algorithm, entropy compression method and the subset gas

Paula M. S. Fialho, Bernardo N. B. de Lima, and Aldo Procacci

Abstract. We establish a connection between the entropy compression method and the Moser–Tardos algorithmic version of the Lovász local lemma through the cluster expansion of the subset gas. We also show that the Moser–Tardos resampling algorithm and the entropy compression bactracking algorithm produce identical bounds.

1. Introduction

1.1. The Lovász local lemma

The Lovász local lemma (LLL), originally formulated by Erdös and Lovász in [19], is a powerful tool in the framework of the probabilistic method used in an impressive quantity of applications in combinatorics such as graph coloring problems, K-sat, latin transversal, etc. (see [5] and references therein for a review). Its basic idea is to prove the existence of some combinatorial object with certain desired properties (e.g., a proper coloring of the vertices of a graph) by identifying a family $\mathfrak{F}$ of *bad events* (also called *flaws*) in some probability space $(\Omega, \mathcal{A}, \mathrm{Prob})$ whose presence, even of only one of them, prevents the object under analysis to occur and whose simultaneous non-occurrence guarantees that the object under analysis is actually present. Denoting by $\bar{e}$ the complement event of $e \in \mathfrak{F}$, the Lovász local lemma provides a condition on the probabilities $\mathrm{Prob}(e)$ in order to ensure that $\mathrm{Prob}(\bigcap_{e\in\mathfrak{F}} \bar{e}) > 0$. To formulate explicitly this condition one need to identify a so-called *dependency graph* for the family $\mathfrak{F}$. That is to say, a graph $\mathcal{G}$ with vertex set $\mathfrak{F}$ and edge set such that each event $e \in \mathfrak{F}$ is independent of the σ-algebra generated by the collection of events $\mathfrak{F} \setminus \Gamma^*_{\mathcal{G}}(e)$, where $\Gamma^*_{\mathcal{G}}(e) = \Gamma_{\mathcal{G}}(e) \cup \{e\}$ and $\Gamma_{\mathcal{G}}(e)$ is the set of all the events of $\mathfrak{F}$ adjacent to e in $\mathcal{G}$. According to the usual terminology, $\Gamma^*_{\mathcal{G}}(e)$ is called the *neighborhood* of e in $\mathcal{G}$ while $\Gamma_{\mathcal{G}}(e)$ is called the *punctured neighborhood* of e in $\mathcal{G}$.

2020 Mathematics Subject Classification. Primary 05D40; Secondary 82B20, 68W20.
Keywords. Lovász local lemma, abstract polymer system, Moser–Tardos algorithm, entropy compression, subset gas.

Once the dependency graph of the family $\mathfrak{F}$ has been determined, the LLL can be stated as follows.

Theorem 1.1 (Lovász local Lemma). *Let $\mathfrak{F}$ be a finite family of events in a probability space $(\Omega, \mathcal{A}, \mathrm{Prob})$ and let $\mathcal{G}$ be a dependency graph for $\mathfrak{F}$. Let $\boldsymbol{\mu} = \{\mu_e\}_{e\in\mathfrak{F}}$ be a collection of non-negative numbers. If, for each $e \in \mathfrak{F}$,*

$$\mathrm{Prob}(e) \le \frac{\mu_e}{\prod_{e'\in\Gamma^*_{\mathcal{G}}(e)}(1+\mu_{e'})}, \tag{1.1}$$

then

$$\mathrm{Prob}\Big(\bigcap_{e\in\mathfrak{F}} \bar{e}\Big) > 0. \tag{1.2}$$

1.2. The abstract polymer gas

The abstract polymer system (APS) is a discrete model originally proposed by Kotecký and Preiss [35] as a generalization of a lattice polymer model introduced by Gruber and Kunz [28] in 1968. Its relevance in statistical mechanics is significant since it is a widely used tool to analyze a large number of systems in physics, such as discrete spin systems, continuous particle systems, percolative processes and even quantum field theories.

The APS is defined by a triple $(\mathcal{P}, \mathbf{w}, W)$ where $\mathcal{P}$ is a countable (possibly infinite) set whose elements are called *polymers,* $\mathbf{w}\colon \mathcal{P} \to \mathbb{C}$ is a function which associates to each polymer $\gamma \in \mathcal{P}$ a complex number w_γ, called the *activity* of the polymer γ, and $W\colon \mathcal{P}\times\mathcal{P} \to \{0,1\}$ is a function called the *Boltzmann factor*, such that $W(\gamma,\gamma) = 0$ and $W(\gamma,\gamma') = W(\gamma',\gamma)$ for all $\{\gamma,\gamma'\} \subset \mathcal{P}$. Usually, the pair $\{\gamma,\gamma'\}$ is called *incompatible* when $W(\gamma,\gamma') = 0$ and *compatible* when $W(\gamma,\gamma') = 1$.

Let $\mathcal{G}$ be the simple graph with vertex set $\mathcal{P}$ and edge set formed by the pairs $\{\gamma,\gamma'\} \subset \mathcal{P}$ such that $W(\gamma,\gamma') = 0$. The graph $\mathcal{G}$, which is uniquely determined by the Boltzmann factor W, is sometimes called the *support graph* of W. The neighborhood of the vertex γ in the graph $\mathcal{G}$ is the set $\Gamma^*_{\mathcal{G}}(\gamma) = \{\gamma' \in \mathcal{P}\colon W(\gamma,\gamma') = 0\}$ formed by all polymers incompatible with γ. An independent set of the support graph $\mathcal{G}$ is a set Y of polymers such that each pair $\{\gamma,\gamma'\} \subset Y$ is compatible. We denote by $I(\mathcal{G})$ the set formed by all finite independent sets of $\mathcal{G}$.

Given a finite collection of polymers $\Lambda \subset \mathcal{P}$, the grand canonical partition function of the APS at "finite volume" Λ is given by

$$Z_\Lambda(\mathbf{w}) = \sum_{\substack{S\subset\Lambda \\ S\in I(\mathcal{G})}} \prod_{\gamma\in S} w_\gamma.$$

This is a key quantity since the thermodynamic properties of the system can be derived from it. In particular, a fundamental question physicists are interested in is to find radii

$\mathbf{R} = \{R_\gamma\}_{\gamma\in\mathcal{P}}$ (with $R_\gamma \geq 0$ for all $\gamma \in \mathcal{P}$) such that the partition function $Z_\Lambda(\mathbf{w})$, for any Λ finite, is free of zeros for all complex activities $\mathbf{w}$ within the polydisk $\{|w_\gamma| < R_\gamma\}_{\gamma\in\mathcal{P}}$ (shortly $\mathbf{w} \leq \mathbf{R}$). This would guarantee that the logarithm of the partition function, which is related to the pressure of the system, is analytic in such regions, so that no phase transition occurs. The best current lower bound for such radii $\mathbf{R}$ is due to Fernández and Procacci [21] who improved the older bounds due to Kotecký and Preiss [35] and Dobrushin [16] proving the following theorem.

Theorem 1.2 (Fernández–Procacci criterion). *Let $\boldsymbol{\mu} = \{\mu_\gamma\}_{\gamma\in\mathcal{P}}$ be a collection of nonnegative numbers such that*

$$|w_\gamma| \leq R_\gamma^{\mathrm{FP}} \equiv \frac{\mu_\gamma}{\Xi_\gamma(\boldsymbol{\mu}, \mathcal{G})} \quad \text{for all } \gamma \in \mathcal{P} \tag{1.3}$$

with

$$\Xi_\gamma(\boldsymbol{\mu}, \mathcal{G}) = \sum_{\substack{S\subseteq\Gamma^*_{\mathcal{G}}(\gamma)\\ S\in I(\mathcal{G})}} \prod_{\gamma'\in S} \mu_{\gamma'}. \tag{1.4}$$

Then, for all finite $\Lambda \subset \mathcal{P}$, $Z_\Lambda(\mathbf{w}) \neq 0$.

The Kotecký–Preiss and the Dobrushin criteria can be formulated analogously with the only difference that function $\Xi_\gamma(\boldsymbol{\mu}, \mathcal{G})$ appearing in the right hand side of (1.3) is replaced respectively by

$$\varphi_\gamma^{\mathrm{KP}}(\boldsymbol{\mu}) = e^{\sum_{\gamma'\in\Gamma^*_{\mathcal{G}}(\gamma)} \mu_{\gamma'}}, \tag{1.5}$$

and

$$\varphi_\gamma^{\mathrm{D}}(\boldsymbol{\mu}) = \sum_{S\subseteq\Gamma^*_{\mathcal{G}}(\gamma)} \prod_{\gamma'\in S} \mu_{\gamma'} = \prod_{\gamma'\in\Gamma^*_{\mathcal{G}}(\gamma)} (1 + \mu_{\gamma'}). \tag{1.6}$$

The bound on radii $\mathbf{R}$ given by (1.3) is always greater than the bounds on the same radii given by the Kotecký–Preiss and the Dobrushin criteria since

$$\exp\Big\{ \sum_{\gamma'\in\Gamma^*_{\mathcal{G}}(\gamma)} \mu_{\gamma'} \Big\} \geq \prod_{\gamma'\in\Gamma^*_{\mathcal{G}}(\gamma)} (1 + \mu_{\gamma'}) = \sum_{S\subseteq\Gamma^*_{\mathcal{G}}(\gamma)} \prod_{\gamma'\in S} \mu_{\gamma'} \geq \sum_{\substack{S\subseteq\Gamma^*_{\mathcal{G}}(\gamma)\\ S\in I(\mathcal{G})}} \prod_{\gamma'\in S} \mu_{\gamma'}.$$

1.3. The connection between the LLL and the APS

The LLL criterion (1.1) is a *sufficient* condition in order to the thesis (1.2) to hold. In 1985, Shearer [47] presented a necessary and sufficient criterion for (1.2) to hold. The Shearer criterion was actually constituted by a set of several conditions which however were very difficult (if not impossible) to verify in practical applications. Probably for this reason Shearer's result went somehow overseen until 2005. In this year Scott

and Sokal [45], revisiting Shearer's work, showed that there was a quite surprising connection between the Lovász Local Lemma and the abstract polymer gas. Scott and Sokal showed that, given the family of events $\mathfrak{F}$ and their dependency graph $\mathcal{G}$, the Shearer criterion is equivalent to require that the probabilities of the bad events, $\{\text{Prob}(e)\}_{e\in\mathfrak{F}}$, fall in the zero-free region of the partition function of the APS whose support graph coincides with the dependency graph $\mathcal{G}$ of the family $\mathfrak{F}$. So, once rephrased in the statistical mechanics lingo, it is no surprise that the Shearer criterion was unusable in practice. On the other hand, Scott and Sokal observed that the APS counterpart of the LLL criterion (1.1) coincides with the aforementioned Dobrushin criterion. Later, Bissacot et al. [12], via the connection disclosed in [45] pointed out that Theorem 1.2 implies the following improvement of the LLL criterion (1.1).

Theorem 1.3 (cluster expansion local lemma (CELL)). *With the same hypothesis of Theorem 1.1, if, for each event* $\mathfrak{e} \in \mathfrak{F}$

$$\text{Prob}(e) \le \frac{\mu_e}{\Xi_e(\boldsymbol{\mu}, \mathcal{G})}, \tag{1.7}$$

with

$$\Xi_e(\boldsymbol{\mu}, \mathcal{G}) = \sum_{\substack{S\subseteq\Gamma^*_{\mathcal{G}}(e)\\ S\in I(\mathcal{G})}} \prod_{e'\in S} \mu_{e'}, \tag{1.8}$$

then

$$\text{Prob}\Big(\bigcap_{e\in\mathfrak{F}} \bar{e}\Big) > 0.$$

As observed above this is clearly an improvement with respect to Theorem 1.1 since

$$\sum_{\substack{S\subseteq\Gamma^*_{\mathcal{G}}(e)\\ S\in I(\mathcal{G})}} \prod_{e'\in S} \mu_{e'} \le \sum_{S\subseteq\Gamma^*_{\mathcal{G}}(e)} \prod_{e'\in S} \mu_{e'} = \prod_{e'\in\Gamma^*_{\mathcal{G}}(e)} (1+\mu_{e'}).$$

Condition (1.7) has been shown to be effective in several applications of the LLL (see, e.g., [12,13,39,42]). This new criterion is nowadays known as "cluster expansion (CE) criterion."

Formulas (1.1) and (1.7) on one hand and formulas (1.3) and (1.6) on the other hand clearly show the evident interchange between the LLL and the APS:

- events $e \in \mathfrak{F}$ in LLL correspond to polymers $\gamma \in \mathcal{P}$ in APS;
- dependent events correspond to incompatible polymers;
- probabilities of events correspond to absolute values of activities of polymers;
- probability of the good event to be positive corresponds to require that the partition function evaluated at $-|\mathbf{w}|$ (this is the worst case, see, e.g., [11, 22, 45]) to be strictly positive.

1.4. Moser–Tardos algorithmic version of the LLL

The LLL is very general, in particular in its statement and proof there is no need to specify anything about the probability space. Of course, in the applications the probability space $(\Omega, \mathcal{A}, \mathrm{Prob})$ has to be specified and it is natural to wonder, once condition (1.1) is satisfied, if it is possible to find a polynomial algorithm in this specified probability space able to find an outcome in the sample space Ω which realizes the event $\bigcap_{e\in\mathfrak{F}} \bar{e}$. During many years, researchers have tried to find methods to devise successful algorithms for as many as possible applications covered by the LLL. These efforts have been only partially successful in the sense that the class of examples for which an efficient algorithm could be found was limited and the condition (1.1) got worse, see for example [4, 9]. Such situation changed radically in 2009, when in a breakthrough paper [38] Moser and Tardos presented a fully algorithmic version of the LLL which covered the vast majority of LLL applications.

The scheme proposed by Moser and Tardos is called nowadays the *variable setting*. It is given a finite set Λ whose elements will be called *atoms* and a collection $\psi_\Lambda \equiv \{\psi_x\}_{x\in\Lambda}$ of mutually independent random variables in a fixed probability space $(\Omega, \mathcal{A}, \mathrm{Prob})$. In general, each random variable ψ_x takes values in its own space Ψ_x according to its own distribution, but in a vast majority of the applications these variables take values in a common finite space Ψ (e.g., the set of colors) according to the uniform distribution. When this occur we say that we are in the so-called *uniform variable setting*. An *assignment* ω_Λ (or *evaluation,* or *realization,* or *configuration*) of values to all variables ψ_Λ is an element $\omega_\Lambda = (\omega_x)_{x\in\Lambda}$ of the Cartesian product $\Omega_\Lambda = \prod_{x\in\Lambda} \Psi_x$. An event $e \in \mathcal{A}$ which depends only on variables of a subset $A \subset \Lambda$ (i.e., e is fully determined by the values assigned to variables $\psi_A = \{\psi_x\}_{x\in A}$) is called henceforth *tempered event* and we write $\mathrm{supp}(e) = A$ and $\mathrm{supp}(e) = A$ is sometimes called the *scope* of e. We also agree that $\Omega_A = \prod_{x\in A} \Psi_x$ and, given $\omega_\Lambda \in \Omega_\Lambda$, $\omega_A = (\omega_x)_{x\in\mathcal{A}}$, and we refer to ω_A as the projection of ω_Λ on A.

Moser and Tardos considered situations in which all of the flaws constituting the family $\mathfrak{F}$ are tempered. With this assumption any two events e, e' of $\mathfrak{F}$ such that $\mathrm{supp}(e) \cap \mathrm{supp}(e') = \emptyset$ are necessarily independent. This implies that the graph $\mathcal{G}$ with vertex set $\mathfrak{F}$ and edge set constituted by the pairs $\{e, e'\}$ such that $\mathrm{supp}(e) \cap \mathrm{supp}(e') \neq \emptyset$ is a natural dependency graph for the family $\mathfrak{F}$.

Example 1. Let $G = (V, E)$ be a graph and let $k \in \mathbb{N}$. A *coloring* of the vertices of G is a function $c\colon V \to [k]$ (where $[k] = \{1, 2, \ldots, k\}$). A coloring c of the vertices of G is *nonrepetitive* if, for any $n \geq 1$, no path $p = \{v_1, v_2, \ldots, v_{2n}\}$ is colored *repetitively,* i.e., such that $c(v_i) = c(v_{i+n})$ for all $i = 1, 2, \ldots, n$. The minimum number of colors needed such that G has a non-repetitive vertex coloring is called the *non-repetitive chromatic index* of G and it is denoted by $\pi(G)$. Here we are in the uniform variable setting, where the set of atoms Λ coincides with the set of vertices V of the graph G

and to each atom/vertex $v \in V$ we associate independently a random variable ψ_v, the *color* of v, that takes values in $[k]$ according to the uniform distribution. Let P_n be the set of all paths with $2n$ vertices and set $P = \bigcup_{n\geq 1} P_n$. The family $\mathfrak{F}$ of bad events is the set $\mathfrak{F} = \{e_p\}_{p\in P}$, where e_p is the event "the path p is colored repetitively" and a coloring avoiding all events in $\mathfrak{F}$ is non-repetitive.

In this setting Moser and Tardos defined the following algorithm.

RESAMPLING

1. Do a random sampling of the variables ψ_Λ and let ω_Λ be the assignment so obtained.
2. While there is a bad event belonging to $\mathfrak{F}$ occurring under ω_Λ, select an occurring event $e \in \mathfrak{F}$ and resample all variables $\{\psi_x\}_{x\in \text{ supp}(e)}$. Let ω'_Λ be the new assignment so produced.
3. End while.
4. Output current.

Moser and Tardos proved that if condition (1.1) of Theorem 1.1 holds, then the algorithm RESAMPLING terminates rapidly finding an assignment $\omega_\Lambda \in \prod_{x\in\Lambda} \Psi_x$ such that none of the bad events of the family $\mathfrak{F}$ occurs. Later, inspired by the paper by Bissacot et al. [12], Pegden [41] improved the Moser–Tardos result replacing condition (1.1) with condition (1.7).

Theorem 1.4 (Pegden). *Given a finite set Λ, let $\{\psi_x\}_{x\in\Lambda}$ be a family of mutually independent random variables in a fixed probability space $(\Omega, \mathcal{A}, \text{Prob})$, and let $\mathfrak{F}$ be a family of tempered bad events with natural dependency graph $\mathcal{G}$. Let $\boldsymbol{\mu} = \{\mu_e\}_{e\in\mathfrak{F}}$ be non-negative numbers. If, for each $e \in \mathfrak{F}$,*

$$\text{Prob}(e) \;\leq\; \frac{\mu_e}{\Xi_e(\boldsymbol{\mu}, \mathcal{G})} \tag{1.9}$$

with $\Xi_e(\boldsymbol{\mu}, \mathcal{G})$ defined in (1.8), then there is an assignment $\omega_\Lambda \in \prod_{x\in\Lambda} \Psi_x$ of the values to variables ψ_Λ avoiding all events from the family $\mathfrak{F}$ and the algorithm RESAMPLING *finds such an assignment ω_Λ in an expected total number of steps less than or equal to $\sum_{e\in\mathfrak{F}} \mu_e$.*

It is also worth to mention that Kolipaka and Szegedy showed in [33] that the algorithm RESAMPLING is successful in polynomial time also if Shearer conditions hold (see also [6] for a similar result).

Remark 1.5. In the Moser–Tardos setting above described the function $\Xi_e(\boldsymbol{\mu}, \mathcal{G})$, defined in (1.8), admits a somehow natural upper bound so that condition (1.9) can be

greatly simplified (paying some price as we will see in a moment). Given $y \in \Lambda$, let us define the set

$$\mathfrak{F}(y) = \{e \in \mathfrak{F} \colon y \in \operatorname{supp}(e)\}. \tag{1.10}$$

Then $\mathfrak{F}(y)$ is a clique of the natural dependency graph $\mathcal{G}$ of the family $\mathfrak{F}$: any pair $\{e, e'\} \subset \mathfrak{F}(y)$ is such that $\operatorname{supp}(e) \cap \operatorname{supp}(e') \supset \{y\} \neq \emptyset$, i.e., any pair $\{e, e'\} \subset \mathfrak{F}(y)$ is an edge of $\mathcal{G}$. Thus, the neighborhood $\Gamma^*_{\mathcal{G}}(e)$ of any event $e \in \mathfrak{F}$ is the union of cliques $\{\mathfrak{F}(y)\}_{y \in \operatorname{supp}(e)}$, namely, $\Gamma^*_{\mathcal{G}}(e) = \bigcup_{y \in \operatorname{supp}(e)} \mathfrak{F}(y)$. Therefore

$$\Xi_e(\boldsymbol{\mu}, \mathcal{G}) \leq \Xi_e^{\text{clique}}(\boldsymbol{\mu}, \mathcal{G}) \equiv \prod_{y \in \operatorname{supp}(e)} \Big[1 + \sum_{e' \in \mathfrak{F}(y)} \mu_{e'}\Big]. \tag{1.11}$$

Using the expression $\Xi_e^{\text{clique}}(\boldsymbol{\mu}, \mathcal{G})$, which is much simpler to evaluate, in place of $\Xi_e(\boldsymbol{\mu}, \mathcal{G})$ in condition (1.7), it was possible to get improved estimates for latin transversal in [12] and improved bounds for several chromatic indices in [13,39]. It is important however to stress that in general the union $\bigcup_{y \in \operatorname{supp}(e)} \mathfrak{F}(y)$ is not disjoint, and when cliques $\{\mathfrak{F}(y)\}_{y \in \operatorname{supp}(e)}$ are too overlapped, the replacement $\Xi_e(\boldsymbol{\mu}, \mathcal{G})$ with $\Xi_e^{\text{clique}}(\boldsymbol{\mu}, \mathcal{G})$ tends to be a too crude estimate. This situation occurs for example in the case of perfect and separating hash families (see [42]).

1.5. The subset gas

In the context of the connection between the LLL and the APS it is natural to ask where the Moser–Tardos setting defined above does fit. We will explain below that the APS counterpart of the Moser–Tardos setting is the so-called subset gas.

The subset gas, originally proposed by Gruber and Kunz [28], is a particular realization of the abstract polymer gas which appears in many physical situations and it is defined as follows.

Given a countable set $\mathbb{V}$, the space of *polymers* $\mathcal{P}$ is defined as the collection of all finite subsets of $\mathbb{V}$, namely $\mathcal{P} = \{\gamma \subset \mathbb{V} \colon |\gamma| < +\infty\}$. The *Boltzmann factor* is then defined as $W(\gamma, \gamma') = 0$ if $\gamma \cap \gamma' \neq \emptyset$ and $W(\gamma, \gamma') = 1$ if $\gamma \cap \gamma' = \emptyset$. Thus, in such a realization of the APS, polymers have a cardinality, so that one can speak about *large* polymers and *small* polymers. Of course, as before, each polymer γ has an associated *activity* w_γ. In most of the physical realizations, $\mathbb{V}$ is the vertex set of a (possibly infinite) graph (typically $\mathbb{V}$ is the cubic lattice $\mathbb{Z}^d$ with the edge set being the set of nearest neighbor in $\mathbb{Z}^d$). Actually, subset gases appearing in the framework of statistical mechanics, specifically spin systems on $\mathbb{Z}^d$, have in general a further characteristic. Commonly, in each site $x \in \mathbb{Z}^d$ is defined a random variable s_x (the *spin* at x) taking values in some space S_x and often this space S_x is the same for all $x \in \mathbb{V}$. Then the space of the polymers $\mathcal{P}$ is formed by the pairs $\gamma = (\operatorname{supp}(\gamma), s_\gamma)$, where $\operatorname{supp}(\gamma)$ is, as before, a finite subset of $\mathbb{V}$ and s_γ is the *spin configuration* of γ,

i.e., a function from $\mathrm{supp}(\gamma)$ to $\prod_{x\in\mathrm{supp}(\gamma)} S_x$. Classical examples are the "thick" contours of the Pirogov–Sinai theory (see, e.g., [48, Chapter II]).

The reader can see at this point the evident parallel with the variable setting of the LLL. Namely, polymers $\gamma = (\mathrm{supp}(\gamma), s_\gamma)$ in the subset gas correspond to (elementary) events e in the Moser–Tardos variable setting.

Generally, as far as the subset gas is concerned, the bound (1.11) is always used. So, for the subset gas, the condition (1.3) can be written as

$$|w_\gamma| \le \frac{\mu_\gamma}{\prod\limits_{x\in\mathrm{supp}(\gamma)} \Big[1 + \sum\limits_{\substack{\gamma'\in\mathcal{P}\\ x\in\mathrm{supp}(\gamma')}} \mu_{\gamma'}\Big]} \quad \text{for all } \gamma \in \mathcal{P}. \tag{1.12}$$

Note that the above criterion is constituted by many inequalities, i.e., as many inequalities as the number of total polymers, so if $\mathcal{P}$ is infinite this number can be infinite. However, in the specific case of the subset gas this set of inequalities can be replaced (and usually is!) by a unique global inequality to which the set of activities must obey. Indeed, since by (1.12) we necessarily have that $\mu_\gamma > |w_\gamma|$, a typical choice is to set

$$\mu_\gamma = |w_\gamma| e^{a|\,\mathrm{supp}(\gamma)|}$$

with $a > 0$. Such a choice permits to resume the set of conditions (1.12) in terms of a simple "global" conditions on the set of activities $\mathbf{w}$. Namely, the thesis of Theorem 1.3 holds if

$$\sup_{x\in\mathbb{V}} \sum_{\substack{\gamma\in\mathcal{P}\\ x\in\mathrm{supp}(\gamma)}} |w_\gamma| e^{a|\,\mathrm{supp}(\gamma)|} \le e^a - 1 \quad \text{for some } a > 0. \tag{1.13}$$

The above discussion on the connection between the LLL and the APS leads to conclude that the set of conditions of the CELL criterion can be reexpressed in terms of a global unique condition of the probabilities $\mathrm{Prob}(e)$ of the events e as far as we are in the Moser–Tardos variable setting.

Lemma 1.6. *Given a finite set Λ, let $\{\psi_x\}_{x\in\Lambda}$ be a family of mutually independent random variables in a fixed probability space $(\Omega, \mathcal{A}, \mathrm{Prob})$, and let $\mathfrak{F}$ be a family of tempered bad events.*

If it is possible to find $a > 0$ such that

$$\sup_{x\in\Lambda} \sum_{\substack{e\in\mathfrak{F}\\ x\in\mathrm{supp}(e)}} \mathrm{Prob}(e) e^{a|\,\mathrm{supp}(e)|} \le e^a - 1, \tag{1.14}$$

then

$$\mathrm{Prob}\Big(\bigcap_{e\in\mathfrak{F}} \bar{e}\Big) > 0.$$

and RESAMPLING *finds an assignment* $\omega_\Lambda \in \bigcap_{e\in\mathfrak{F}} \bar{e}$ *of variables* $\{\psi_x\}_{x\in\Lambda}$ *in an expected total number of steps less than or equal to* $\sum_{e\in\mathfrak{F}} \mathrm{Prob}(e)e^{a|\,\mathrm{supp}(e)|}$.

The latter global "subset gas condition" (1.14) has been used in [12, 13, 39]) to reproduce all improvements obtained via CELL with the exception of [42], where the general condition (1.7) has been used. It is worth to mention that similar (but less effective) global conditions deduced from the original LLL in the variable setting have been already formulated in the literature and used in specific examples (see, e.g., [36, Lemma 3] and reference therein).

1.6. The entropy compression method

The Moser–Tardos algorithmic version of the LLL, since its appearance, has been the subject of a very intense study by several researchers in the areas of computer science, combinatorics, and probability. In this regard, two main directions can be pointed out. The first one concerns the (successful) efforts made to extend the validity of the algorithmic version of the LLL beyond the variable setting (see, e.g., [1,2,30–32,34], and references therein) with the motivation to include important applications of the non-constructive LLL (such as latin transversal) which does not fit in the independent variable setting.

The second direction was motivated by the hope that the algorithm RESAMPLING proposed by Moser and Tardos, being extremely simple, could be reasonably modified/refined in order to improve the final criterion beyond (1.7). These ideas have been originally developed in [18, 20, 29] where backtracking algorithms have been implemented for specific graph coloring problems to obtain bounds which are better than those obtainable by LLL or CELL.

In particular, Esperet and Parreau devised in [20] an algorithm able to obtain a new upper bound for the chromatic index of the acyclic edge coloring of a graph with maximum degree Δ which sensibly improves on the bound obtained by Ndreca et al. [39] just an year before via the CELL. The algorithm proposed by Esperet and Parreau presents evident differences from the algorithm RESAMPLING: instead of sampling all variables at once and then resampling some of them until all flaws are avoided, the Esperet–Parreau algorithm starts by assuming that in the beginning no variable has an assigned value. Then, step by step, a (random) value is attributed to a currently unassigned variable; when this leads to the appearance of one or more flaws, the algorithm backtracks to a partial non-violating assignment by withdrawing the values of some set of variables. To prove that their algorithm could find in a polynomial time a configuration of the variables that avoids all bad events, Esperet and Parreau basically used a counting argument, therefore tacitly assuming that the random variables were taking values in a common finite set according to the uniform distribution.

Esperet and Parreau suggested, through examples and applications, that their algorithm could be adapted to treat most of the applications in graph coloring problems covered by the LLL. Indeed, this was confirmed in several successive papers [8, 14, 15, 25, 27, 40, 43, 44, 46], where the Esperet–Parreau scheme was applied to various graph coloring problems and beyond, generally improving previous results obtained via the LLL/CELL (sometimes the improvement is more noticeable, sometimes less). However, in all papers mentioned above, the Esperet–Parreau algorithmic scheme, usually called *entropy compression method* (the name is probably due to Tao [49]), has been commonly utilized as a set of *ad hoc* instructions to be implemented on a case-by-case basis. A systematization of the entropy compression method providing a general criterion similar to those given by the LLL and CELL has been an open question since the beginning ([20,27]) and it is still demanded even in the very recent paper by Achlioptas and Iliopoulos [3]. In this regard we mention a non-algorithmic general criterion proposed by Bernshteyn [10] which is able to reproduce several results obtained by the entropy compression method. The systematization of the Esperet–Parreau backtrack algorithm can actually be found in [7] in which the setting where entropy compression can be used is clearly outlined, a general *entropy compression criterion* is proposed and a connection between bounds obtained via this method and LLL conditions is elucidated.

According to [7], the entropy compression method can be implemented in any application that can be analyzed through a variable setting *a la* Moser–Tardos with the further restriction that, for all $x \in \Lambda$, variables ψ_x take values in a common space $\Psi_x = [k]$, where $k \in \mathbb{N}$ (or, more generally, in possibly distinct spaces Ψ_x, but all with the same cardinality k) according to the uniform distribution. We will refer to this particular realization of the variable setting as the *uniform variable setting*. Let us review rapidly, for later comparison, the entropy compression criterion obtained in [7].

1.6.1. The entropy compression setting. As said above, the entropy compression method can be applied in the so-called *uniform variable setting* where all random variables $\{\psi_x\}_{x\in\Lambda}$ take values in the common space $\Psi = [k] \equiv \{1, 2, \dots, k\}$ according to the uniform distribution. In this case we can work in very simple finite product probability space. Namely, we can suppose that *all* events in this space are uniquely determined by the values of variables $\{\psi_x\}_{x\in\Lambda}$. We thus may assume that the sample space coincides with the set of all possible assignment of the variables $\{\psi_x\}_\Lambda$, i.e., we take as sample space the product space $\Omega_\Lambda = [k]^\Lambda$ in such a way that the variables $\{\psi_x\}_{x\in\Lambda}$ are just projections, namely, $\psi_x \colon \Omega_\Lambda \to \Psi_x \colon \omega_\Lambda \mapsto \omega_x$, and an assignment ω_Λ of all the variables coincides with an outcome in the sample space. The event space $\mathcal{A}$ is then the power set of Ω_Λ, namely $\mathcal{A} = 2^{\Omega_\Lambda}$, and thus all events in this space are by construction tempered. Finally, Prob is the product probability measure induced

by the uniform distribution of each variable on $[k]$. That is to say, for any $\omega_\Lambda \in \Omega_\Lambda$, we have that $\mathrm{Prob}(\omega_\Lambda) = 1/k^{|\Lambda|}$ and for any $e \in 2^\Omega$, we have that $\mathrm{Prob}(e) = |e|/k^{|\Lambda}$ where $|e|$ is the number of outcomes forming e. This (finite product) probability space is completely determined by the pair (Λ, k) and will be denoted below with the symbol $\mathcal{S}_{\Lambda,k}$. Of course, by construction, in $\mathcal{S}_{\Lambda,k}$ the variables ψ_Λ are mutually independent. We sometimes refer to $[k]$ as the set of "colors" and, by the above definitions, saying that the variable ψ_x has an assigned value is the same as saying that is colored.

Borrowing the terms from the statistical mechanics lingo and from graph theory, we will also refer to an outcome $\omega \in [k]^\Lambda$ in the probability space $\mathcal{S}_{\Lambda,k}$ (i.e., an assignment of the variables $\{\psi_x\}_{x\in\Lambda}$) either as a *configuration* of $[k]^\Lambda$ or as a *coloring* of the atoms of Λ.

In this framework, given $A \subset \Lambda$ and an event e in $\mathcal{S}_{\Lambda,k}$ with $\mathrm{supp}(e) = A$, we set $e^* = \{\omega_A \in [k]^A : \omega_\Lambda \in e\}$ so that $|e| = k^{|\Lambda|-|\mathrm{supp}(e)|}|e^*|$. The event e is *elementary* if $|e|/k^{|\Lambda|-|\mathrm{supp}(e)|} = |e^*| = 1$.

Moreover, given a family $\mathfrak{F}$ of bad events of $\mathcal{S}_{\Lambda,k}$, a *good configuration with respect to* $\mathfrak{F}$ is a configuration $\omega_\Lambda \in [k]^\Lambda$ avoiding all events in $\mathfrak{F}$; that is to say, ω_Λ is such that, for all $e \in \mathfrak{F}$, we have $\omega_{\mathrm{supp}(e)} \notin e^*$. Given $e \in \mathfrak{F}$ and $X \subsetneq \mathrm{supp}(e)$, we set $e|_X = \{w_X \in [k]^X : w \in e\}$ and we call $e|_X$ the *restriction* of the event e on X (so in particular $e|_{\mathrm{supp}(e)} = e^*$).

Remark 1.7. More generally, one can also suppose that variables $\{\psi_x\}_{x\in\Lambda}$ take values in possibly different spaces $\{\Psi_x\}_{x\in\Lambda}$ but all having a common cardinality k, i.e., such that $|\Psi_x| = k$ for all $x \in \Lambda$, and in each space Ψ_x the random variable ψ_x takes values according to the uniform distribution. Example 2 in Section 4 below falls in this more general setting.

Remark. We stress once again that virtually all applications of the entropy compression method available in the literature fall in this uniform variable setting, with the sole exception, as far as we know, of the acyclic edge coloring of bounded degree graphs (see [7]).

Definition 1.8. In the entropy compression setting determined by the pair (Λ, k), given an event e, a non-empty subset $X \subset \mathrm{supp}(e)$ is called a *seed* of e if $e|_X = [k]^X$ and $|e^*| = k^{|X|}$. An event e is said *tidy* if either it is elementary, or it is such that for all $y \in \mathrm{supp}(e)$, there exists a non-empty set $X \subset \{\mathrm{supp}(e) \setminus \{y\}\}$ which is a seed of e.

Clearly, by definition, all seeds of a tidy event e must have all the same cardinality which we denote by $\kappa(e)$. If e is elementary, we set $\kappa(e) = 0$. We further set

$$\|e\| = |\,\mathrm{supp}(e)| - \kappa(e) \tag{1.15}$$

and refer to $\|\mathfrak{e}\|$ as *the power of the event* $\mathfrak{e}$. The properties listed here below hold.

1. If e is tidy and X is a seed of e, then any configuration $\omega_{\mathrm{supp}(e)} \in e^*$ is uniquely determined by its projection ω_X on X and no $Y \subsetneq X$ has this property.
2. If e is tidy, then
$$\mathrm{Prob}(e) = \frac{1}{k^{|\,\mathrm{supp}(e)|-\kappa(e)}}.$$
3. Let $\mathfrak{F}$ be a family of tempered and tidy events. Suppose that an event e is occurring in a given configuration and $X \subsetneq \mathrm{supp}(e)$ is a seed of e. If we resample all the variables in $\mathrm{supp}(e) \setminus X$ leaving all other variables unchanged, then in the so obtained new configuration the probability for an event $e' \in \mathfrak{F}$ to occur is $\mathrm{Prob}(e')$.

All these properties follows immediately from Definition 1.8 of seed. Property (1) is self-evident. Property (2) follows by observing that $k^{\kappa(e)}k^{|\Lambda|-|\,\mathrm{supp}(e)|}$ is the number of favorable configurations for the event e while $k^{|\Lambda|}$ is the number of total configurations. Concerning property (3), observe that, if a configuration ω_Λ is given in which we know that the event e with seed X occurs, since $\{\psi_x\}_{x\in\Lambda}$ are mutually independent, the projection $\omega_{\Lambda\setminus\mathrm{supp}(e)}$ of ω_Λ on $\Lambda \setminus \mathrm{supp}(e)$ can be anyone of the possible outcomes in $[k]^{\Lambda\setminus\mathrm{supp}(e)}$ and, since X is seed of e, also the projection ω_X of ω_Λ on X could be any configuration in $[k]^X$. Therefore, by resampling anew the variables $\{\psi_x\}_{x\in\mathrm{supp}(e)\setminus X}$ we fall in a new configuration ω'_Λ in which the variables $\{\psi_x\}_{x\in\mathrm{supp}(e)}$ can assume any of the possible assignments in $[k]^{\mathrm{supp}(e)}$ and therefore the new configuration ω'_Λ so obtained could be any configuration of $[k]^\Lambda$.

Remark 1.9. In [7] the definition of seed and tidy event is slightly more general. Namely, the definition of seed coincides with property (1) in the above list: a seed $X \subset \{\mathrm{supp}(e) \setminus \{y\}\}$ is such that any configuration in e^* is uniquely determined by its projection on X and no $Y \subsetneq X$ has this property. The difference is subtle. As an example, suppose that the edges of a graph $G = (V, E)$ are colored at random using k colors, uniformly and independently and consider a cycle C of G with an even number of edges. Let e be the event "C is properly bichromatic," i.e., e is the event in which the edges of C are colored with two colors and no pair of adjacent edges is monochromatic. Then any cherry (i.e., a pair of adjacent edges) of C is a seed of e according to the definition given in [7], while e is not tidy according to Definition 1.8, since, for any cherry $c \in C$ the restriction $e|_c$ to c is not the whole $[k]^c$, once the monochromatic configurations of c are not allowed if C is properly bichromatic. On the other hand, considering Example 1 introduced in Section 1.4 about non-repetitive vertex colorings of a graph $G = (V, E)$, if $p \in P$ is a path in G with $2n$ vertices, then either the first half of p or the second half of p are seeds of the event $\mathfrak{e}_p = \{\text{the path } p \text{ is colored repetitively}\}$. Thus, $\mathfrak{e}_p$ is tidy with seeds of size n and therefore $\|\mathfrak{e}_p\| = n$ and $\mathrm{Prob}(\mathfrak{e}_p) = \frac{1}{k^n}$.

Remark 1.10. Note that, if e is not tidy, then it can be seen as the disjoint union of elementary (hence tidy) events. Therefore, there is no loss of generality in considering only families in which all events are tidy.

Recalling (1.10), we set

$$\mathfrak{F}_s(y) = \{\mathrm{e} \in \mathfrak{F}(y) \colon \|\mathrm{e}\| = s\} \tag{1.16}$$

and

$$d_s = \max_{y \in \Lambda} |\mathfrak{F}_s(y)|. \tag{1.17}$$

In particular, d_s is an upper bound for the number of events with power s whose support contains a common element of Λ. We finally define

$$E'_{\mathfrak{F}} = \{s \in \mathbb{N} \colon \text{there exists } \mathrm{e} \in \mathfrak{F} \text{ such that } \|\mathrm{e}\| = s\}. \tag{1.18}$$

To fix the ideas, the reader may consider once again Example 1 introduced in Section 1.4 concerning non-repetitive vertex colorings of a graph $G = (V, E)$. Clearly, if $p \in P$ is a path in G with $2n$ vertices, either the first half of p or the second half of p are seeds of the event $\mathrm{e}_p = \{\text{the path } p \text{ is colored repetitively}\}$. Thus, e_p is tidy with seeds of size n and therefore $\|\mathrm{e}_p\| = n$ and $\mathrm{Prob}(\mathrm{e}_p) = \frac{1}{k^n}$. So, in this case $E'_{\mathfrak{F}} = \{1, 2, 3, \dots\}$ and d_s coincides with the maximum number of paths in G of size $2s$ containing a fixed vertex which, if Δ is the maximum degree of G, can be bounded as $d_s \leq s\Delta^{2s-1}$.

1.6.2. The entropy compression algorithm and the entropy compression lemma. Setting $[k]_0 = [k] \cup \{0\}$, a *partial configuration (or partial assignment)* w is a function $w \colon \Lambda \to [k]_0$ and when $w(x) = 0$ we say that the variable ψ_x is *unassigned* (or *uncolored*). We assume that a total order has been chosen in the sets Λ and $\mathfrak{F}$. We choose, for each $y \in \Lambda$ and $\mathrm{e} \in \mathfrak{F}(y)$, a unique subset $S_y(\mathrm{e}) \subset \mathrm{supp}(\mathrm{e}) \setminus \{y\}$ such that $S_y(\mathrm{e})$ is a seed of e. We also denote shortly $S^c_y(\mathrm{e}) = \mathrm{supp}(\mathrm{e}) \setminus S_y(\mathrm{e})$. Note that $y \in S^c_y(\mathrm{e})$. Given a partial configuration $w \colon \Lambda \to [k]_0$, given $X \subset \Lambda$ and given $s \in [k] \cup \{0\}$, we denote by $w]^s_X$ the partial configuration which coincides with w in the set $\Lambda \setminus X$ and it takes the value s at every $x \in X$. If $X = \{x\}$, we set shortly $\omega]^s_{\{x\}} \equiv \omega]^s_x$. If $s \in [k]_0$, we denote by $w^{-1}(s)$ the inverse image of the set $\{s\}$ under the function w. Namely, $w^{-1}(s) = \{x \in \Lambda \colon w(x) = s\}$.

Let t be an arbitrary natural number (which can be taken as large as we please) and let V_t be an element of $[k]^t$, i.e., V_t is a vector with t entries such that each entry takes values in the set $[k]$.

The algorithm ENTROPY COMPRESSION has input V_t, performs (at most) t steps and, at each step $i \in [t]$, produces a partial coloring w_i as described below.

ENTROPY COMPRESSION (with input V_t)

Step 0.
Set $w_0 = 0$, i.e., in the beginning no variable ψ_x, with $x \in \Lambda$, is assigned.

Step i (for $i \geq 1$).

i°. If $w_{i-1}^{-1}(0) \neq \emptyset$, let y be the smallest element of Λ (in the total order chosen) such that ψ_y is unassigned in the partial coloring w_{i-1}. Take the i-th entry of the vector V_t and let $s \in [k]$ be this entry. Assign to ψ_y the value s and consider the partial configuration $w_{i-1}]_y^s$ so obtained.

i_1°. If no event of the family $\mathfrak{F}$ occurs under $w_{i-1}]_y^s$, set $w_i = w_{i-1}]_y^s$ and go to the step $i + 1$.

i_2°. Conversely, if some events in the family $\mathfrak{F}$ occur in $w_{i-1}]_y^s$, select the smallest, say e, which by construction belongs to the set $\mathfrak{F}(y)$. Set $w_i = w_{i-1}]_{S_y^c(e)}^0$ and go to the step $i + 1$. In words, w_i is obtained from w_{i-1} by discoloring all ψ_x such that $x \in \mathrm{supp}(e) \setminus S_y(e)$.

$i^\bullet$. If $w_{i-1}^{-1}(0) = \emptyset$, stop the algorithm discarding all entries $v_i, v_{i+1}, \ldots, v_t$ of V_t.

Note that the partial coloring w_i returned by the algorithm at the end of each step i necessarily avoids all flaws in $\mathfrak{F}$. ENTROPY COMPRESSION performs at most t steps but it can stop earlier, i.e., after having performed $m < t$ steps and $w_m^{-1}(0) = \emptyset$. In this case, only the first m entries of the vector V_t are used. ENTROPY COMPRESSION is successful if it stops after $m < t$ steps, or it lasts t steps and after the last step t we have $\omega_t^{-1}(0) = \emptyset$. Conversely, ENTROPY COMPRESSION fails if it performs all t steps and $\omega_t^{-1}(0) \neq \emptyset$. Clearly, when ENTROPY COMPRESSION is successful w_t is a good configuration. Observe that ENTROPY COMPRESSION can be either deterministic, if V_t is a given prefixed vector, or random, if the entries of V_t are uniformly sampled from the set $[k]$ sequentially and independently. In [7] the following theorem is proved.

Theorem 1.11 (entropy compression lemma). *Assume that a pair (Λ, k) is given together with a family $\mathfrak{F}$ of tidy events in $\mathcal{S}_{\Lambda,k}$. If there is $\alpha > 0$ such that*

$$\frac{1}{\alpha}\Big(1 + \sum_{s \in E'_{\mathfrak{F}}} d_s \alpha^s\Big) < k, \tag{1.19}$$

then

$$\bigcap_{e \in \mathfrak{F}} \bar{e} \neq \emptyset.$$

Moreover, ENTROPY COMPRESSION *finds a configuration* $\omega_\Lambda \in \bigcap_{e\in\mathfrak{F}} \bar{e}$ *in an expected number of steps linear in* $|\Lambda|$.

We will refer to the inequality (1.19) as the *entropy compression criterion*. We stress that this theorem is able to reproduce all results obtained in these last years via the entropy compression method.

It is now simple to compare condition (1.19) with the global CELL criterion (1.14). Since we are in the entropy compression setting, the probability space is the space $\mathcal{S}_{\Lambda,k}$ described above and thus, for any $e \in \mathfrak{F}$, we have $\mathrm{Prob}(e) = \frac{1}{k^{\|e\|}}$. Setting

$$q = \max_{e\in\mathfrak{F}} \left\{ \frac{|\operatorname{supp}(e)|}{\|e\|} \right\} \tag{1.20}$$

and grouping events in terms of their powers we can bound the right hand side of (1.14) as

$$\sup_{x\in\Lambda} \sum_{\substack{e\in\mathfrak{F} \\ x\in\operatorname{supp}(e)}} \mathrm{Prob}(e) e^{a|\operatorname{supp}(e)|} \leq \sum_{s\in E'_{\mathfrak{F}}} \frac{d_s}{k^s} e^{aqs}.$$

So, condition (1.14) is fulfilled if

$$\sum_{s\in E'_{\mathfrak{F}}} d_s \left(\frac{e^{aq}}{k} \right)^s \leq e^a - 1, \tag{1.21}$$

or, setting $\alpha = \frac{e^{aq}}{k}$, if there is $\alpha > 0$ such that

$$\frac{1}{\alpha} \left(1 + \sum_{s\in E'_{\mathfrak{F}}} d_s \alpha^s \right)^q \leq k. \tag{1.22}$$

The reader can immediately compare (1.22) with the entropy compression condition (1.19). The presence of the exponent q defined in (1.20) in inequality (1.22) is the only reason why entropy compression condition (1.19) can give better bounds than LLL. It must however be stressed that we are excluding here the case of the acyclic edge coloring of a graph with maximal degree Δ. In this peculiar case, the entropy compression scheme is fruitfully combined with the crucial observation that it is possible to properly color the edges of G using just $2\Delta - 1$ color in such a way to avoid bichromatic cycles of length 4 (see in [7, Lemma 4]). This observation leaded Esperet and Parreau to a very noticeable improvement of the upper bound of the acyclic edge chromatic index of G with respect to the bound obtained via CELL. Due to its specificity, we have considered the case of the acyclic edge coloring in a separate paper [23] (see also comments below and see also the remark in [7, Section 4.2.5]).

1.7. Motivations and plan of this paper

Concluding this introduction, we need to mention two recent papers, [26,37], proposing a variant of the Moser–Tardos resampling algorithm, which has actually motivated the present work. In particular, in [26] Giotis et al. are able to improve precisely the special case of the acyclic edge chromatic index of a graph. The intriguing fact is that Giotis et al. use in [26] the Moser–Tardos resampling algorithm with the unique variant that the successive resampled bad events must be chosen, when possible, in the neighbor of the previous bad event. Their result is somehow surprising considering that, as mentioned above, the CELL criterion applied to acyclic edge coloring gives a much worse bound than the entropy compression method.

In the present paper, we manage to combine the ideas of [26] (also foreshadowed in [37]) with the observation explained above that the power of a tidy event can be considered in place of the cardinality of its support and we show that the criterion (1.19) based on the backtracking algorithm ENTROPY COMPRESSION can be reobtained in the usual Moser–Tardos scheme by doing the slight modification of the algorithm RESAMPLING illustrated in [26, 37] (which give rise to forests instead of trees as registers of the steps of the algorithm) jointly with the prescription proposed by Esperet–Parreau to not resample certain variables of the bad events (the previously seen "seeds" of the events).

The rest of the paper is organized as follows. In Section 2 we describe the variant of the Moser–Tardos algorithm and state our main result, i.e., Theorem 2.2. Section 3 is devoted to the proof of Theorem 2.2. Finally, in Section 4 we present some examples.

2. A variant of the Moser–Tardos algorithm

Let us go back to the general Moser–Tardos setting described in Section 1.4. We have a finite set Λ (with cardinality $m \equiv |\Lambda|$) and a family $\psi_\Lambda \equiv \{\psi_x\}_{x\in\Lambda}$ of m mutually independent random variables is some probability space $(\Omega, \mathcal{A}, \mathrm{Prob})$ such that each ψ_x has range Ψ_x.

Analogously to the entropy-compression case, an assignment of variables ψ_Λ is a configuration $\omega_\Lambda \in \Omega_\Lambda = \prod_{x\in\Lambda} \Psi_x$ and, according to Moser and Tardos, we consider only tempered events of $\mathcal{A}$. We recall that in this general setting an event $e \in \mathcal{A}$ is tempered if e depends only on variables of a subset $A \subset \Lambda$ (i.e., e is fully determined by variables $\psi_A = \{\psi_x\}_{x\in A}$). As before, we write $\mathrm{supp}(e) = A$.

Let us consider a finite collection $\mathfrak{F}$ of tempered events in the probability space $(\Omega, \mathcal{A}, \mathrm{Prob})$.

Definition 2.1 (seed). Let $e \in \mathfrak{F}$. A proper subset $U \subsetneq \mathrm{supp}(e)$ is called a *seed* of e if it is such that, given a configuration $\omega_\Lambda \in \Omega_\Lambda$ which produces the event e, if we resample all the variables in $\mathrm{supp}(e) \setminus U$ leaving unchanged the values of all the other variables, then in the new configuration ω'_Λ so obtained all the events $e' \in \mathfrak{F}$ have at most $\mathrm{Prob}(e')$ to happen and any $U' \supset U$ has not this property. An event $e \in \mathfrak{F}$ is *tidy* if it is such that all seeds of e have the same non-zero cardinality $\kappa(e)$ and for all $x \in \mathrm{supp}(e)$ there exists a seed $U \subsetneq \mathrm{supp}(e)$ of e such that $x \notin U$.

Remark. Of course, in the uniform variable setting described in Section 1.6.1, where the variables $\{\psi_x\}_{x\in\Lambda}$ beside being independents are also identically and uniformly distributed and taking values in the common finite set $[k] = \{1, \dots, k\}$, Definition 2.1 and Definition 1.8 are equivalent. Note moreover that, according to Definition 2.1, the transition $\omega_\Lambda \to \omega'_\Lambda$, which does not increase the probability of any event in $\mathfrak{F}$, reminds the definition of *resampling oracle* given in [32].

As previously, we define the *power* of the event e as the number

$$\|e\| = \begin{cases} |\,\mathrm{supp}(e)| - \kappa(e) & \text{if } e \text{ is tidy,} \\ |\,\mathrm{supp}(e)| & \text{otherwise.} \end{cases} \tag{2.1}$$

Moreover, for any tidy event $e \in \mathfrak{F}$ and any $x \in \mathrm{supp}(e)$, we fix a rule to choose uniquely a seed $S_x(e)$ of e such that $x \notin S_x(e)$.

As we did in Section 1.6.1, we classify the events $e \in \mathfrak{F}$ according to their power $\|e\|$. So, $E'_{\mathfrak{F}} \subset \mathbb{N}$ is defined as in (1.18), i.e.,

$$E'_{\mathfrak{F}} = \{s \in \mathbb{N} : \text{there exists } e \in \mathfrak{F} \text{ such that } \|e\| = s\}. \tag{2.2}$$

Moreover, as in (1.17), for $s \in E'_{\mathfrak{F}}$, we set

$$d_s = \max_{y\in\Lambda} \left|\mathfrak{F}_s(y)\right| \tag{2.3}$$

Finally, we define

$$p_s = \max_{e\in\mathfrak{F}_s} \mathrm{Prob}(e), \tag{2.4}$$

where $\mathfrak{F}_s = \{e \in \mathfrak{F} : \|e\| = s\}$.

Following [26], we now describe a procedure, called FOREST ALGORITHM which samples (and eventually resamples) the variables ψ_Λ. Given an assignment ω_Λ of all variables ψ_Λ, we say shortly that the atom $x \in \Lambda$ is *bad* if some $e \in \mathfrak{F}$ occurs under ω_Λ and $x \in \mathrm{supp}(e)$. Otherwise, we say that x is *good*. We are also agree here below that to sample the variables ψ_Λ means to pick a random configuration ω_Λ in Ω_Λ.

FOREST ALGORITHM

1. Sample all variables ψ_Λ.
2. While there is a bad atom, select the pair (x, e) where x is the smallest bad atom and e is the smallest event occurring such that $x \in \text{supp}(e)$, and do
3. RESAMPLE(x, e).
4. End while.
5. Output current evaluation.

RESAMPLE(x, e)

1. Resample all variables $\{\psi_y\}$ such that $y \in \text{supp}(e) \setminus S_x(e)$.
2. While there is a bad atom in $\text{supp}(e) \setminus S_x(e)$, let x' be the smallest of these atoms and let e' be the smallest event occurring such that $x' \in \text{supp}(e')$ and do
3. RESAMPLE(x', e').
4. End while.

A *step* of the FOREST ALGORITHM is the procedure described in Line 2 of RESAMPLE(x, e). Since $x \notin S_x(e)$ for any e such that $x \in \text{supp}(e)$, in RESAMPLE(x, e) the variable ψ_x is always resampled. A *phase* of the FOREST ALGORITHM is the collection of steps made by the FOREST ALGORITHM during a call of RESAMPLE(x, e) in Line 3 of the FOREST ALGORITHM. Note that during a phase many steps occur, the first step of the i-th phase will be called the *root* of the phase i. The *record* of the algorithm is the sequence

$$\mathcal{L} = ((x_1, e_1), (x_2, e_2), \dots)$$

constituted by the steps done by the algorithm during its execution. We will denote by *atom label* (resp. *event label*) any atom (resp. event) listed in the record $\mathcal{L}$. According to the prescriptions described above, $\mathcal{L}$ is a random variable determined by the random samplings performed by the algorithm in each step. If $\mathcal{L}$ is finite, i.e., if $|\mathcal{L}| = n$ for some $n \in \mathbb{N}$, then the algorithm terminates having performed n steps and produces an outcome $\omega_\Lambda \in \bigcap_{e \in \mathfrak{F}} \bar{e}$. Let us define

$$P_n = \text{Prob}(|\mathcal{L}| \geq n). \tag{2.5}$$

In other words P_n is the probability that the FOREST ALGORITHM runs at least n steps.

We are now in the position to state the main result of this paper. Recalling definitions (2.2), (2.3), and (2.4), for $s \in E'_{\mathfrak{F}}$ and $\xi > 0$, let

$$\phi_{\mathfrak{F}}(\xi) = \sum_{s \in E'_{\mathfrak{F}}} p_s d_s (\xi + 1)^s. \tag{2.6}$$

Theorem 2.2. *Given a finite set Λ, let $\{\psi_x\}_{x \in \Lambda}$ be a family of mutually independent random variables in a fixed probability space $(\Omega, \mathcal{A}, \mathrm{Prob})$, and let $\mathfrak{F}$ be a family of tempered bad events in this space. Suppose that*

$$\min_{\xi > 0} \frac{\phi_{\mathfrak{F}}(\xi)}{\xi} < 1, \tag{2.7}$$

then there is an assignment ω_Λ of values to the variables ψ_Λ such that none of the events in the family $\mathfrak{F}$ occur. Moreover, the FOREST ALGORITHM *finds such an assignment in an expected number of steps polynomial in $m = |\Lambda|$.*

Remark. Note that (2.7) is equivalent to the condition (1.19) of the entropy compression lemma (Theorem 1.11). Indeed, in the uniform variable setting defined by the pair (Λ, k) where the probability space is the finite product space $\mathcal{S}_{\Lambda,k}$ defined in Section 1.6.1, we have that $p_s = \frac{1}{k^s}$ and therefore, posing $\alpha = (\xi + 1)/k$, (2.7) is rewritten in the form (1.19).

3. Proof of Theorem 2.2

Let us start by proving some important properties of the FOREST ALGORITHM.

Lemma 3.1. *Consider any call of* RESAMPLE(x, e) *and let Y be the set of all good atoms at the beginning of this call. If this call finishes, then all the atoms in $Y \cup \{\mathrm{supp}(\mathrm{e}) \setminus S_x(\mathrm{e})\}$ are good.*

Proof. According to the algorithm, if RESAMPLE(x, e) finishes, then $\mathrm{supp}(\mathrm{e}) \setminus S_x(\mathrm{e})$ are good atoms, so we just need to prove that the atoms in Y continue to be good in the end of RESAMPLE(x, e). Let $y \in Y$, and assume that RESAMPLE(x, e) finishes and performs n steps. Suppose by contradiction that after these n steps performed by RESAMPLE(x, e) y is bad. Then there exists a last step $t \leq n$ of RESAMPLE(x, e) such that y was good at step $t - 1$, became bad at step t, and stayed bad during the remaining $n - t$ steps of RESAMPLE(x, e). This means that there is an event e' and an atom $z \in \mathrm{supp}(\mathrm{e}')$ such that RESAMPLE(z, e') was called at step $t - 1$ and $y \in \mathrm{supp}(\mathrm{e}') \setminus S_z(\mathrm{e}')$ became bad as soon as the variables $\psi_{\mathrm{supp}(\mathrm{e}') \setminus S_z(\mathrm{e}')}$ were resampled. But RESAMPLE(z, e') must end at a step $t' > t$ and at this step all variables of $\mathrm{supp}(\mathrm{e}') \setminus S_z(\mathrm{e}')$ must be good and thus y, which belongs to $\mathrm{supp}(\mathrm{e}') \setminus S_z(\mathrm{e}')$, is good at step $t' > t$ in contradiction with the assumption. ∎

Lemma 3.2. *The* FOREST ALGORITHM *performs at most* $m = |\Lambda|$ *phases.*

Proof. Consider two phases l and s, with $l < s$, generated by an execution of the FOREST ALGORITHM and let (x_l, e_l) and (x_s, e_s) be the pairs resampled at their initial steps respectively, i.e., the roots of phase l and s respectively. By Lemma 3.1, all atoms in $\mathrm{supp}(e_l) \setminus S_{x_l}(e_l)$ are good when phase l ends and at the beginning of any successive phase. In particular, since $x_l \in \{\mathrm{supp}(e_l) \setminus S_{x_1}(e_l)\}$, x_l is good and thus $x_l \notin \mathrm{supp}(e_s)$. In conclusion $x_l \neq x_s$. ∎

3.1. Witness forest

We will associate to an execution of the FOREST ALGORITHM a labeled forest formed by plane rooted trees whose vertices are labeled with pairs (x, e) belonging to $\mathcal{L}$.

Suppose that the algorithm performs r phases and during the phase s, with $s \in \{1, \dots, r\}$, the algorithm performs n_s steps, in such a way that the record of the algorithm is

$$\mathcal{L} = ((x_1^1, e_1^1), \dots, (x_{n_1}^1, e_{n_1}^1), (x_1^2, e_1^2), \dots, (x_{n_2}^2, e_{n_2}^2), \dots, (x_1^r, e_1^r), \dots, (x_{n_r}^r, e_{n_r}^r)). \tag{3.1}$$

At each phase s, $1 \le s \le r$, we will associate a tree τ'_s. Let

$$(x_1^s, e_1^s), \dots, (x_i^s, e_i^s), \dots, (x_{n_s}^s, e_{n_s}^s), \tag{3.2}$$

be the pairs resampled at phase s. We construct the tree τ'_s in the following way.

a) The root of τ'_s has label (x_1^s, e_1^s).

b) For $i > 1$, we proceed by checking if (x_i^s, e_i^s) is such that

$$x_i^s \in (\mathrm{supp}(e_{i-1}^s) \setminus S_{x_{i-1}^s}(e_{i-1}^s)):$$

- if yes, we add (x_i^s, e_i^s) as a child of (x_{i-1}^s, e_{i-1}^s);
- if no, we go back in (3.2) checking the *ancestors* of the vertex labeled by (x_{i-1}^s, e_{i-1}^s) until we find a pair (x_j^s, e_j^s), with $j < i$, such that $x_i^s \in (\mathrm{supp}(e_j^s) \setminus S_{x_j^s}(e_j^s))$, and we add (x_i^s, e_i^s) as a child of (x_j^s, e_j^s).

Observe that by the construction of the FOREST ALGORITHM, all pairs (x_i^s, e_i^s) can be added to τ'_s in this way, then τ'_s has n_s vertices (leaves included) with labels (x_i^s, e_i^s) with $i = 1, \dots, n_s$. By Lemma 3.1 the pair (x_1^{s+1}, e_1^{s+1}) is the first pair in (3.1) that cannot be added to τ'_s in this way, so we build a new tree τ'_{s+1} with root (x_1^{s+1}, e_1^{s+1}) following the same rule described to build τ'_s.

Note that the vertices of the forest defined above are naturally ordered according to the natural order of the steps made by the algorithm. The forest $F' = \{\tau'_1, \dots, \tau'_r\}$ so

obtained uniquely associated to the record $\mathcal{L}$ is such that, for each $s \in [r]$, τ'_s is a rooted plane tree with n_s vertices and each vertex of τ'_s has label (x, e) where $x \in \mathrm{supp}(e)$ and $e \in \mathfrak{F}$.

Note that, by Lemma 3.2, we have that $r \leq m$ and thus the forest F' contains at most m trees.

Note also that in each tree τ'_s of F' the list of labels of the vertices of τ'_s ordered according to the depth-first search, coincides with the list (3.2).

Note finally that, by construction, the correspondence $\mathcal{L} \mapsto F'$ is an injection.

Example 2. Let $G = (V, E)$ be a graph with $|V| = 600$. Suppose that we want a coloring of the vertices of G that is nonrepetitive (see Example 1). For any path $p \in P$ with an even number of vertices, recall that e_p is the event "the path p is colored repetitively." Given a vertex $v \in p$, if v belongs to the first half of the path p, the seed $S_v(e_p)$ is the second half of the path and conversely, if v belongs to the second half of the path p, the seed $S_v(e_p)$ is the first half of the path. Suppose that the vertices of G have been colored following the FOREST ALGORITHM and the record of the algorithm after five steps is the following:

$$\mathcal{L} = ((v_1, e_{p_1}), (v_2, e_{p_2}), (v_2, e_{p_2}), (v_4, e_{p_4}), (v_9, e_{p_9})), \tag{3.3}$$

where the paths p_1, p_2, p_4 and p_9 are formed by the vertices

$$\begin{aligned} p_1 &= v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, \\ p_2 &= v_2, v_8, v_{16}, v_{32}, v_{64}, v_{128}, v_{256}, v_{512}, \\ p_4 &= v_4, v_6, v_9, v_{12}, \\ p_9 &= v_9, v_{18}, v_{27}, v_{36}, v_{45}, v_{54}, v_{63}, v_{72}, v_{81}, v_{90}. \end{aligned}$$

Let us construct the forest F' associated to the record $\mathcal{L}$ given in (3.3). Since $\mathcal{L}$ is formed by five steps, this forest will have five nodes, call them u_1, u_2, u_3, u_4, u_5. The first node u_1, necessarily the root of the first tree of the forest, has label (v_1, e_{p_1}), the first entry of $\mathcal{L}$. Observe that v_2 (the vertex of the second entry of $\mathcal{L}$) is a vertex in the path p_1 such that $v_2 \in \mathrm{supp}(e_{p_1}) \setminus S_{v_1}(e_{p_1})$, then the second node u_2 of the forest F' with label by (v_2, e_{p_2}) is a child of u_1. Let us then consider the third entry of $\mathcal{L}$, observe that v_2 is in $\mathrm{supp}(e_{p_2}) \setminus S_{v_2}(e_{p_2})$, so the third node u_3 of the forest F', labeled by (v_2, e_{p_2}) is a child of u_2. The fourth entry of $\mathcal{L}$ is the pair (v_4, e_{p_4}), observe that $v_4 \notin \mathrm{supp}(e_{p_2}) \setminus S_{v_2}(e_{p_2})$, so the fourth node u_4 of the forest, with (v_4, e_{p_4}) cannot be a child of u_3. By the same reason u_4 cannot be a child of u_2. However, as $v_4 \in \mathrm{supp}(e_{p_1}) \setminus S_{v_1}(e_{p_1})$, we fix the node u_4 labeled by (v_4, e_{p_4}) as a child of u_1.

Consider now the last entry of $\mathcal{L}$, as $v_9 \notin \mathrm{supp}(e_{p_4}) \setminus S_{v_4}(e_{p_4})$, once $S_{v_4}(e_{p_4}) = \{v_9, v_{12}\}$, we have that (v_9, e_{p_9}) cannot be a child of u_4. We also have that $v_9 \notin \mathrm{supp}(e_{p_1}) \setminus S_{v_1}(e_{p_1})$, therefore u_5 cannot be a child of the node u_1. This means that

u_1, u_2, u_3, u_4 are the nodes of the first tree τ'_1 of the forest F' while u_5 with label (v_9, e_{p_9}) is an isolated node constituting the second tree τ'_2 of the forest F'.

Then we have produced the forest $F' = \{\tau'_1, \tau'_2\}$, represented in Figure 1, that contains a total of $|\mathfrak{L}| = 5$ nodes.

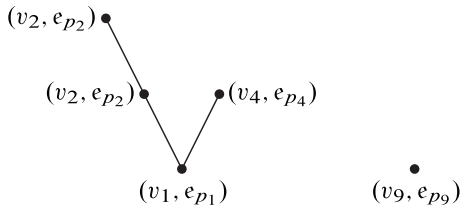

Figure 1. Forest F' associated to the record $\mathfrak{L}$ given in (3.3).

Lemma 3.3. *Consider a tree $\tau' \in F'$, and let v_i and v_j be two vertices in τ' with labels (x_i, e_i) and (x_j, e_j) respectively. We have that*

a. *if v_i is a child of v_j, then $x_i \in \mathrm{supp}(\mathrm{e}_j) \setminus S_{x_j}(\mathrm{e}_j)$;*

b. *if v_i and v_j are siblings in τ', then $x_i \neq x_j$.*

c. *any vertex $v \in \tau'$ with label (x, e) has at most $\|\mathrm{e}\|$ children, where $\|\mathrm{e}\|$ is defined in (2.1).*

Proof. a. It is trivial by construction of the algorithm.

b. As v_i and v_j are siblings, suppose that v_i and v_j are the i-th and the j-th children of a vertex in τ', with $i < j$ in the natural order of the vertices of τ' induced by the steps of the algorithm. For q such that $i \leq q < j$, let (x_q, e_q) be the label of the q-th sibling. By Lemma 3.1 when RESAMPLE(x_q, e_q) ends all the atoms $x_i, \ldots, x_q$ are good as well the atoms in $(\mathrm{supp}(\mathrm{e}_i) \setminus S_{x_i}(\mathrm{e}_i)) \cup \cdots \cup (\mathrm{supp}(\mathrm{e}_q) \setminus S_{x_q}(\mathrm{e}_q))$. Therefore, x_j cannot be in the set $\{x_i, x_{i+1}, \ldots, x_{j-1}\}$.

c. Follows trivially from items (a) and (b). ∎

Given a forest F' produced by the algorithm, we let $X_{F'}$ be the set of atoms which label the roots of the trees of the forest, i.e.,

$$X_{F'} = \{x \in \Lambda : \text{there exists } \mathrm{e} \in \mathfrak{F} \text{ such that } (x, \mathrm{e}) \text{ is the root label of some } \tau' \in F'\}.$$

Lemma 3.2 implies that atoms in $X_{F'}$ are all distinct.

Definition 3.4 (witness forest). Given the record $\mathfrak{L}$ of the FOREST ALGORITHM and the forest F' associate to $\mathfrak{L}$, we construct a new forest F by adding to F' new vertices in the following way.

1. Add to the forest F' as many isolated vertices as the atoms which are in $\Lambda \setminus X_{F'}$, and give to these isolated vertices the label $(x, \emptyset)$ for all $x \in \Lambda \setminus X_{F'}$.
2. for each vertex v of the forest F' with label (x, e) with less than $\|e\|$ children, do the following. Let H_v be the set of atoms in $\mathrm{supp}(e) \setminus S_x(e)$ which are not atoms labels of the children of v. For each $y \in H_v$ we add to v a leaf with label $(y, \emptyset)$ in such a way that v has now exactly $\|e\|$ children.

The new labeled forest F, so obtained uniquely associated to the random variable $\mathcal{L}$ by the prescriptions described above, is called the *witness forest* produced by the FOREST ALGORITHM.

In Figure 2 we show the witness forest obtained from the forest F' of Figure 1, generated by the sequence (3.3) in Example 2. The nodes labeled by $(v_i, \emptyset)$ (the empty circles), added following the rules of Definition 3.4, are leaves of the witness forest F. The nodes labeled by (v_i, e_{p_i}) (the black circles), i.e., the original nodes of F', are now the internal nodes of the witness forest F.

The witness forest F has, by construction, the following properties.

Definition 1 (properties of the witness forest F). 1. F is constituted by exactly $|\Lambda| = m$ labeled rooted trees $\tau_1, \dots, \tau_m$ (some of which are just isolated vertices).

2. Let the vertex u be a child of the vertex v in $\tau \in F$ and let (x_u, e_u) and (x_v, e_v) be their labels respectively. Then $x_u \in \mathrm{supp}(e_v)$.

3. Each internal vertex v of $\tau \in F$ carries a label (x_v, e_v), where $x_v \in \mathrm{supp}(e_v)$ and $e_v \in \mathfrak{F}$, while each leaf ℓ of τ carries a label $(x_\ell, \emptyset)$ and $x_\ell \in \mathrm{supp}(e_w)$, where w is the father of ℓ.

4. Let the vertices v and v' be the i-th and the j-th siblings in $\tau \in F$, with $i < j$ in the depth-first search order of τ, and let (x_i, e_i) and (x_j, e_j) be their labels respectively, then $x_i \neq x_j$.

5. Let v be an internal vertex of $\tau \in F$ and let (x, e) be its label, then the vertex v has exactly $\|e\|$ children.

Let $\mathcal{F}_n$ be the set of labeled forests satisfying properties (1)–(5) above that contains n internal vertices in total and let $\mathcal{F} = \bigcup_{n \geq 0} \mathcal{F}_n$.

It is important to stress that the map $\mathcal{L} \mapsto F$ is an injection. Therefore, since the FOREST ALGORITHM lasts n steps if and only if the witness forest associated to the record $\mathcal{L}$ of the FOREST ALGORITHM has n internal vertices, the probability P_n defined in (2.5) can be written as

$$P_n = \mathrm{Prob}(\text{the witness forest associated to } \mathcal{L} \text{ has } n \text{ internal vertices}). \tag{3.4}$$

Then the next goal is to estimate the probability that the FOREST ALGORITHM produces a witness forest F with n internal vertices.

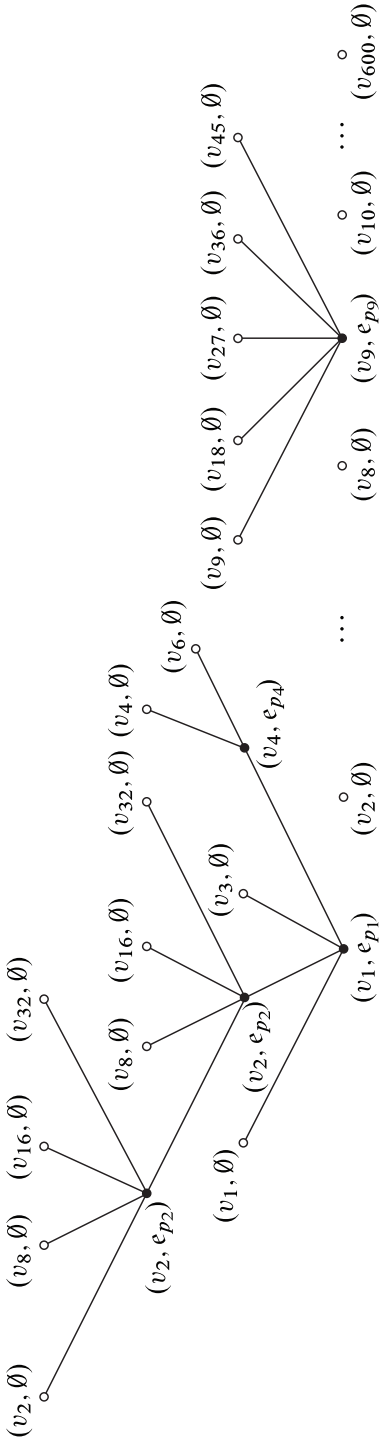

Figure 2. Witness forest F obtained from the forest F' of Figure 1.

3.2. The validation algorithm

Definition 3.5 (admissible sequence). We say that a sequence

$$S = \{(x_1, e_1), \ldots, (x_n, e_n)\}$$

is *admissible* if $x_i \in \mathrm{supp}(e_i)$, for all $i = 1, \ldots, n$.

Given a witness forest F with n internal vertices, we can associate to F, in a natural way, the admissible sequence $S_F = \{(x_1, e_1), \ldots, (x_n, e_n)\}$ formed by the labels of its internal vertices. Namely, the sequence S_F coincides with (3.1).

We now describe a validation algorithm, called S-CHECK, whose input is an admissible sequence $S = \{(x_1, e_1), \ldots, (x_n, e_n)\}$. S-CHECK first samples all variables in ψ_Λ and then resamples some of the variables in ψ_Λ.

S-CHECK
Given the admissible sequence $S = \{(x_1, e_1), \ldots, (x_n, e_n)\}$

1. Sample all variables in ψ_Λ.
2. For $i = 1, \ldots, n$, do
3. If e_i occurs, resample all the variables ψ_y with $y \in \mathrm{supp}(e_i) \setminus S_{x_i}(e_i)$. If the event e_i does not occur, return *failure*.
4. End for.

The procedure described at Line 3 of S-CHECK is called a *step*. Of course, if $S = \{(x_1, e_1), \ldots, (x_n, e_n)\}$ is the input for S-CHECK, its execution will perform exactly n steps if it does not return failure. Observe that S-CHECK does not return failure (i.e., passes) if, and only if, in each step i the event e_i occurs under the current evaluation of the variables.

Lemma 3.6. *Let $S = \{(x_1, e_1), \ldots, (x_n, e_n)\}$ be an admissible sequence. Then*

$$\mathrm{Prob}(S\text{-CHECK } \textit{with input } S \textit{ passes}) \leq \prod_{i=1}^{n} \mathrm{Prob}(e_i). \tag{3.5}$$

Proof. Consider the first step of S-CHECK: we sample all variables ψ_Λ reaching a configuration ω_0 and we have to check if the event e_1 happens; if e_1 does not happen we stop, otherwise we resample the variables in $\mathrm{supp}(e_1) \setminus S_{x_1}(e_1)$. As $S_{x_1}(e_1)$ is a seed, we have that the new configuration ω_1 is such that any event e has probability to occur at most $\mathrm{Prob}(e)$. Therefore, by induction, at each step i the probability of e_i to occur is at most $\mathrm{Prob}(e_i)$. As S-CHECK is successful if and only if all events e_i occur, then inequality (3.5) follows. ∎

Lemma 3.7. *Given a witness forest $F \in \mathcal{F}_n$ whose internal vertices carry labels*

$$S_F = \{(x_1, e_1), \dots, (x_n, e_n)\},$$

we have that

$$\text{Prob}(\text{FOREST ALGORITHM } \textit{produces } F) \leq \prod_{i=1}^{n} \text{Prob}(e_i). \tag{3.6}$$

Proof. Observe that if all the random choices made by an execution of the FOREST ALGORITHM that produces F as witness forest are also made by the algorithm S-CHECK with input S_F, then in each step i the event e_i occurs and so S-CHECK does not return failure. Then

$$\text{Prob}(\text{FOREST ALGORITHM produces } F) \leq \text{Prob}(S_F\text{-CHECK with input } S_F \text{ passes}). \tag{3.7}$$

Now, (3.6) follows from Lemma 3.6 ∎

Remark. In the i-th phase of the algorithm S-CHECK just the values of the variables $\psi_{\text{supp}(e_i)\setminus S_{x_i}(e_i)}$ are resampled. So, at the beginning of phase $i+1$ the distribution of the evaluation of the variables is the same as the Line 1. This is not the case for the FOREST ALGORITHM, because otherwise this would mean that the algorithm is not making any progress in its search of a configuration such that none of event in $\mathfrak{F}$ occurs.

3.3. The unlabeled forest

The strategy to prove Theorem 2.2 is to show that the probability that the FOREST ALGORITHM lasts at least n steps decays exponentially in n, which implies that the FOREST ALGORITHM terminates almost surely, returning an evaluation of ψ_Λ such that all events in $\mathfrak{F}$ do not occur.

If the FOREST ALGORITHM lasts n steps, then it produces a witness forest with n internal nodes. Recall that, if an internal node v of the witness forest has event label e_v, then this node has exactly $s_v = \|e_v\|$ children.

Let $\mathcal{F}_n^*$ be the set of all unlabeled forests constituted by $|\Lambda| = m$ plane trees having in total n internal vertices and such that each internal vertex v has a number of children in the set $E'_{\mathfrak{F}}$ defined in (2.2). Given the record $\mathcal{L}$ of the FOREST ALGORITHM such that $|\mathcal{L}| = n$ and given the witness forest $F \in \mathcal{F}_n$ associated to $\mathcal{L}$, we define the function

$$f : \mathcal{F}_n \to \mathcal{F}_n^*, \quad F \mapsto f(F) = \Phi$$

that removes all the labels of F obtaining an unlabeled witness forest $\Phi \in \mathcal{F}_n^*$. We call Φ the *unlabeled witness forest* produced by the FOREST ALGORITHM.

Figure 3 here below represents the unlabeled witness Φ forest obtained from the witness forest F from Figure 2. The nodes represented by empty circles are the leaves of Φ and the black circles are the internal vertices of Φ.

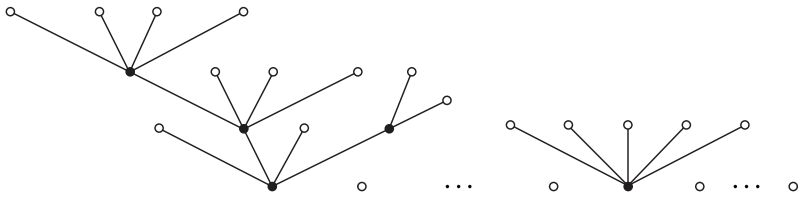

Figure 3. The unlabeled witness forest Φ associated to the witness forest F of Figure 2.

For $\Phi \in \mathcal{F}_n^*$ let us define

$$P_\Phi = \text{Prob}(\text{FOREST ALGORITHM produces the unlabeled witness forest } \Phi). \quad (3.8)$$

Given $F \in \mathcal{F}_n$ and $\Phi \in \mathcal{F}_n^*$, let V_F and V_Φ be the set of internal vertices of F and Φ respectively. Then we have that

$$P_\Phi = \sum_{\substack{F \in \mathcal{F}_n \\ f(F)=\Phi}} \text{Prob}(\text{FOREST ALGORITHM produces the witness forest } F) \quad (3.9)$$

$$\leq \sum_{\substack{F \in \mathcal{F}_n \\ f(F)=\Phi}} \prod_{v \in V_F} \text{Prob}(e_v) \quad (3.10)$$

$$\leq \prod_{v \in V_\Phi} p_{s_v} \sum_{\substack{F \in \mathcal{F}_n \\ f(F)=\Phi}} 1, \quad (3.11)$$

where inequality (3.10) is due to Lemma 3.7 and in (3.11) s_v denotes the number of children of the vertex v, and p_s is defined in (2.4). Recalling now the definition of d_s given in (2.3), we have that

$$\sum_{\substack{F \in \mathcal{F}_n \\ f(F)=\Phi}} 1 \leq \prod_{v \in V_\Phi} d_{s_v}, \quad (3.12)$$

Indeed, for each vertex $v \in V_\Phi$ with s_v children, we have d_{s_v} options for its event label, and, fixed the atom label and the event label of the father of v, we can determine uniquely the atom label of v. Suppose that the father of v is the vertex u, which has s_u children, and x_u is its atom label, then for the event label of u we have d_{s_u} options, and once fixed the event label of u, suppose e_u, we know the atom labels of all children

of u, namely they are, in order, the atoms in $\mathrm{supp}(e_u) \setminus S_{x_u}(e_u)$. So, if v is the i-th child of u, then its atom label is the i-th atom in $\mathrm{supp}(e_u) \setminus S_{x_u}(e_u)$. Proceeding recursively, observe that what we need to know is the atom label of the roots of each tree in Φ; however this information is easily obtained by the construction of a witness forest, as the trees are organized by the atom labels of their roots.

In conclusion,

$$P_\Phi \le \prod_{v \in V_\Phi} d_{s_v} p_{s_v}. \tag{3.13}$$

We now can bound the probability P_n (see (3.4)) that the FOREST ALGORITHM lasts at least n steps as

$$P_n \le \sum_{\Phi \in \mathcal{F}_n^*} P_\Phi.$$

To estimate $\sum_{\Phi \in \mathcal{F}_n^*} P_\Phi$, observe that every forest $\Phi \in \mathcal{F}_n^*$ is constituted by m trees $\tau_1, \dots, \tau_m$ with $n_1, \dots, n_m$ internal vertices respectively. The numbers $n_1, \dots, n_m$ are such that $n_i \ge 0$ for all $i = 1, \dots, m$ and $n_1 + n_2 + \cdots + n_m = n$. Recall also that the number of children of the internal vertices of any τ_i takes values in the set $E'_{\mathfrak{F}}$. Let us denote by $\mathcal{T}$ the set of plane trees with number of children of the internal vertices taking values in the set $E'_{\mathfrak{F}}$ and let $\mathcal{T}_n$ be the subset of $\mathcal{T}$ formed by the trees with exactly n internal vertices.

Let us denote shortly, for $s \in E'_{\mathfrak{F}}$,

$$w_s = d_s p_s. \tag{3.14}$$

For a tree $\tau \in \mathcal{T}$, let V_τ be the set of its internal vertices. Then define the weight of τ as

$$w(\tau) = \prod_{v \in V_\tau} w_{s_v}$$

where we recall that s_v is the number of children of the vertex v.

For a given $n \in \mathbb{N}$, let

$$Q_n = \sum_{\tau \in \mathcal{T}_n} w(\tau).$$

Therefore, the probability that the FOREST ALGORITHM lasts at least n steps is bounded by

$$P_n \le \sum_{\substack{n_1 + \cdots + n_m = n \\ n_i \ge 0}} Q_{n_1} \dots Q_{n_m}. \tag{3.15}$$

It is now easy to check that Q_n is defined by the recurrence relation

$$Q_n = \sum_{s \in E'_{\mathfrak{F}}} w_s \sum_{\substack{n_1 + \cdots + n_s = n-1 \\ n_1 \ge 0, \dots, n_s \ge 0}} Q_{n_1} \dots Q_{n_{k-l}}, \tag{3.16}$$

with $Q_0 = 1$. Now, let

$$W(z) = \sum_{n=1}^{\infty} Q_n z^n,$$

be the generating function encoding the sequence $\{Q_n\}_{n\geq 1}$. Then we have from (3.16)

$$W(z) = z \sum_{n=1}^{\infty} \sum_{s\in E'_{\mathfrak{F}}} w_s \sum_{\substack{n_1+\cdots+n_s=n-1 \\ n_1\geq 0,\ldots,n_s\geq 0}} Q_{n_1} z^{n_1} \ldots Q_{n_s} z^{n_s}, \tag{3.17}$$

$$= z \sum_{s\in E'_{\mathfrak{F}}} w_s \prod_{i=1}^{s} \sum_{n_i\geq 0} Q_{n_i} z^{n_i}, \tag{3.18}$$

$$= z \sum_{s\in E'_{\mathfrak{F}}} w_s \prod_{i=1}^{s} \Big[1 + \sum_{n_i\geq 1} Q_{n_i} z^{n_i}\Big], \tag{3.19}$$

$$= z \sum_{s\in E'_{\mathfrak{F}}} w_s (1 + W(z))^s, \tag{3.20}$$

i.e., denoting, for $\xi > 0$,

$$\phi_{\mathfrak{F}}(\xi) = \sum_{s\in E'_{\mathfrak{F}}} w_s (1+\xi)^s \tag{3.21}$$

we have

$$W(z) = z\phi_{\mathfrak{F}}(W(z)) \tag{3.22}$$

and thus, by a well-known result in analytic combinatorics (see, e.g., [24, Proposition IV.5] or also [17, Theorem 5]), we have that the coefficients of the generating function $W(z)$ are bounded as follows:

$$Q_n \leq \rho^n, \tag{3.23}$$

where

$$\rho = \min_{\xi>0} \frac{\phi_{\mathfrak{F}}(\xi)}{\xi}. \tag{3.24}$$

Hence,

$$P_n \leq \sum_{\substack{n_1+\cdots+n_m=n \\ n_i\geq 0}} Q_{n_1} \ldots Q_{n_m} \leq \rho^n \sum_{\substack{n_1+\cdots+n_m=n \\ n_i\geq 0}} 1 = \rho^n \binom{n+m-1}{m-1} \tag{3.25}$$

Now, if (2.7) holds, we have that the probability that the FOREST ALGORITHM runs at least n steps decays exponentially in n if n is sufficiently large. In particular, it is easy to check that

$$\rho^n \binom{n+m-1}{m-1} \leq \rho^{\frac{n}{2}}$$

as soon as

$$\frac{n}{\ln n} \geq \frac{2m}{|\ln(\rho)|},$$

i.e., as soon as

$$n \geq \frac{2m}{|\ln(\rho)|} \ln^2\left(\frac{2m}{|\ln(\rho)|}\right) \equiv N.$$

Thus, if we estimate $P_n = 1$ if $n \leq N$ and $P_n \leq \rho^{n/2}$ if $n > N$, the expected number of steps T of the FOREST ALGORITHM is given by

$$T \leq \frac{N(N+1)}{2} + \sum_{n=N+1}^{\infty} n\rho^{\frac{n}{2}}.$$

4. Examples

In what follows $G = (V, E)$ is a graph with maximum degree Δ and $k \in \mathbb{N}$. A *coloring* of the vertices (resp. edges) of G is a function $c\colon V \to [k]$ (resp. $c'\colon E \to [k]$).

Nonrepetitive vertex coloring of a graph. We start by revisiting Example 1 presented in Section 1.4.

We recall that in this case we are in the uniform variable setting, where the set of atoms Λ coincides with V and to each atom/vertex $v \in V$ we associate a random variable ψ_v, the color of v, that takes values in $[k]$ according to the uniform distribution. We have set P_n to be the set of all paths with $2n$ vertices and $P = \bigcup_{n\geq 1} P_n$. In this example, the family $\mathfrak{F}$ of bad events is the set $\mathfrak{F} = \{e_p\}_{p\in P}$, where e_p is the event "the path p is colored repetitively." For any $p \in P$, given a vertex v, a seed of e_p not containing v is the half of p that does not contain v. Thus, if p is a path with $2n$ vertices, then e_p is tidy with seeds of size n and therefore $\|e_p\| = n$ and $\mathrm{Prob}(e_p) = \frac{1}{k^n}$. So, in this case,

$$E'_{\mathfrak{F}} = \{1, 2, 3, \dots\}.$$

In order to apply Theorem 2.2, we have to estimate d_s, the maximum number of events of power s containing a fixed vertex. In the present case, d_s coincides with the maximum number of paths in G of size $2s$ containing a fixed vertex. Given that Δ is the maximum degree of G, we have

$$d_s \leq s\Delta^{2s-1}.$$

Therefore, the function $\phi_{\mathfrak{F}}(\xi)$ defined in (2.6) is in the present case

$$\begin{aligned}\phi_{\mathfrak{F}}(\xi) &= \sum_{s\geq 1} s\Delta^{2s-1}\frac{1}{k^s}(\xi+1)^s \\ &= \frac{1}{\Delta}\sum_{s\geq 1} s\Big(\frac{\Delta^2}{k}(\xi+1)\Big)^s \\ &= \frac{1}{\Delta}\frac{\frac{\Delta^2}{k}(\xi+1)}{\Big(1-\frac{\Delta^2}{k}(\xi+1)\Big)^2}, \\ &= \frac{1}{\Delta}\frac{(b+1)(\xi+1)}{(b-\xi)^2},\end{aligned}$$

where in the last line we have set

$$k = (1+b)\Delta^2.$$

Thus, condition (2.7) is, in this case,

$$\min_{\xi>0}\Big(\frac{1}{\Delta}\frac{(b+1)(\xi+1)}{\xi(b-\xi)^2}\Big) < 1.$$

Observe that the minimum occurs at

$$\xi_0 = \frac{\sqrt{9+8b}-3}{4},$$

and

$$\frac{\phi_{\mathfrak{F}}(\xi_0)}{\xi_0} = \frac{1}{\Delta}\frac{\sqrt{(8b+9)^3}+8b^2+36b+27}{8b^3},$$

and thus, if we let $b_0(\Delta)$ be the solution of the equation

$$\frac{\sqrt{(8b+9)^3}+8b^2+36b+27}{8b^3} = \Delta,$$

then we have that the non-repetitive chromatic index $\pi(G)$ of a graph with maximum degree Δ is such that

$$\pi(G) \leq (1+b_0(\Delta))\Delta^2. \tag{4.1}$$

Comparing our bound with [27, Theorem 8], which states that

$$\pi(G) \leq \Delta^2 + \Delta^{\frac{3}{2}}\Big[\frac{3}{2^{2/3}} + \frac{2^{2/3}}{\Delta^{\frac{1}{3}}-2^{\frac{1}{3}}}\Big], \tag{4.2}$$

we observe that the bound (4.1) is better than (4.2) for low values of Δ, while it becomes asymptotically equivalent for large values of Δ.

Facial Thue choice index of planar graphs. We suppose here that the graph $G = (V, E)$ is planar. Suppose moreover that for all edge $e \in E$, a list L_e of k colors is given. A *facial path* of G is a path of G which is part of the boundary of a face of G. The least integer k such that for every collection of lists $\{L_e\}_{e\in E}$ with $|L_e| = k$ there is an edge coloring of G such that every facial path of G is nonrepetitive is called the *facial Thue choice index* of G and is denoted by $\pi'_{fl}(G)$. Observe that the set of independent random variables is in this case $\Psi = \{L_e\}_{e\in E}$.

Let P denotes the set of all facial paths with even number of edges. For all $p \in P$ let e_p be the event "p is repetitive," i.e., if $p = \{e_1, \dots, e_n, e_{n+1}, \dots, e_{2n}\}$, then we have $c'(e_i) = c'(e_{i+n})$ for all $i \in [n]$, where $c'(e)$ is the color chosen in the list L_e via the random experiment. The family of bad events is thus $\mathfrak{F} = \{\mathrm{e}_p\}_{p\in P}$. Observe that, analogously to the previous example, any event e_p with $p \in \mathcal{P}$ is tidy and we can take as a seed of e_p the first or the second half of the path p. This implies that as before $E'_{\mathfrak{F}} = \{1, 2, 3, \dots, \}$. Moreover, if $|p| = 2n$, we have that $\mathrm{Prob}(\mathrm{e}_p) \le \binom{1}{k^n}$, and since every edge of a planar graph is contained in at most $4n$ facial paths of G of size $2n$, for $s \in E'_{\mathfrak{F}}$ we have that $d_s \le 4s$, and therefore

$$\phi_{\mathfrak{F}}(\xi) \le \sum_{s\ge 1} \binom{1}{k^s} 4s(\xi+1)^s < \frac{4^{\frac{\xi+1}{k}}}{\left(1 - \binom{\xi+1}{k}\right)^2} = \frac{4k(\xi+1)}{(k-\xi-1)^2}.$$

Then, we have

$$\min_{\xi>0} \frac{\phi_{\mathfrak{F}}(\xi)}{\xi} < 1$$

as soon as $k \ge 12$, which is the same bound obtained in [43] via entropy compression method.

Coloring graphs frugally. A proper vertex coloring of a graph G is said β-*frugal* if any vertex has at most β members of any color class in its neighborhood. The minimum number of colors required such that a graph G has at least one β-frugal proper vertex coloring is called the β-frugal chromatic number of G and will be denoted by $\chi_\beta(G)$. Analogously to Example 1, we are in the entropy compression setting where $(\Lambda, k) \equiv (V, k)$ and to each $v \in V$ we associate a random variable ψ_v (the color of v) that takes values in $[k]$ according to the uniform distribution.

Observe that in the present case we have only two kind of bad events. First, the coloring has to be proper. So, for each edge $e = \{u, v\}$ of G, we must avoid the event e_e that "u and v have the same color," and let $\mathfrak{F}_1 = \{\mathrm{e}_e\}_{e\in E}$.

We say that a set σ formed by $\beta + 1$ vertices of G is a β-*star* of G if all members of σ are neighbors of a common vertex $v \in V$; in other words, if there is $v \in V$ such that $\sigma \subset \Gamma_G(v)$. Let S_β denote the set of all β-stars of G. Given $\sigma \in S_\beta$, let e_σ be the event "all the $\beta + 1$ vertices forming σ receive the same color," i.e., σ is

monochromatic. We thus have a second family of bad events $\mathfrak{F}_\beta = \{e_\sigma\}_{\sigma\in S_\beta}$. Clearly, the events of the family $\mathfrak{F}_1$ are tidy with seeds of size 1 and power equal to 1, while all events of the family $\mathfrak{F}_\beta$ are tidy with seeds of size 1 and power equal to β. A β-frugal coloring of the vertices of G occurs if none of the events of the family $\mathfrak{F} = \mathfrak{F}_1 \cup \mathfrak{F}_\beta$ occurs.

In the present case, $E'_{\mathfrak{F}} = \{1, \beta\}$, and for every $e \in E$ and $\sigma \in S_\beta$, we have that $\mathrm{Prob}(e_e) = \frac{1}{k}$ and $\mathrm{Prob}(\sigma) \le \frac{1}{k^\beta}$ respectively. To check condition (2.7) we just need to estimate d_s. Observe that $d_1 = \Delta$ and

$$d_\beta \le \Delta\binom{\Delta}{\beta} \le \frac{\Delta^{1+\beta}}{\beta!}.$$

Then, in the present case the function $\phi_{\mathfrak{F}}(\xi)$ defined in (2.6) takes the form

$$\phi_{\mathfrak{F}}(\xi) \le \frac{\Delta}{k}(\xi+1) + \frac{1}{k^\beta}\frac{\Delta^{1+\beta}}{\beta!}(\xi+1)^\beta.$$

Hence, with some calculation, we obtain the upper bound

$$\chi_\beta(G) \le \frac{\Delta^{1+\frac{1}{\beta}}}{\beta!^{1/\beta}}\beta(\beta-1)^{\frac{1}{\beta}-1} + \Delta,$$

which, of course, is the same bound obtained in [7] via entropy compression method.

Acknowledgement. We thank an anonymous referee for valuable comments and suggestions.

Funding. All authors are partially supported by the Brazilian science foundations Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG).

References

[1] D. Achlioptas and F. Iliopoulos, Random walks that find perfect objects and the Lovász local lemma. *J. ACM* **63** (2016), no. 3, Article id. 22 Zbl 1426.05154 MR 3549532

[2] D. Achlioptas, F. Iliopoulos, and V. Kolmogorov, A local lemma for focused stochastic algorithms. *SIAM J. Comput.* **48** (2019), no. 5, 1583–1602 Zbl 07128125 MR 4025776

[3] D. Achlioptas, F. Iliopoulos, and A. Sinclair, Beyond the Lovász local lemma: point to set correlations and their algorithmic applications. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science*, pp. 725–744, IEEE Comput. Soc. Press, Los Alamitos, CA, 2019 MR 4228196

[4] N. Alon, A parallel algorithmic version of the local lemma. *Random Structures Algorithms* **2** (1991), no. 4, 367–378 Zbl 0768.05086 MR 1125955

[5] N. Alon and J. H. Spencer, *The probabilistic method*. Fourth edn., Wiley-Intersci. Ser. Discrete Math. Optim, John Wiley & Sons, Hoboken, N.J., 2016 Zbl 1333.05001 MR 3524748

[6] R. G. Alves and A. Procacci, Witness trees in the Moser–Tardos algorithmic Lovász local lemma and Penrose trees in the hard-core lattice gas. *J. Stat. Phys.* **156** (2014), no. 5, 877–895 Zbl 1302.60021 MR 3231046

[7] R. G. Alves, A. Procacci, and R. Sanchis, A local lemma via entropy compression. 2019, arXiv:1509.04638v2

[8] T. Bartnicki, B. Bosek, S. Czerwiński, M. Farnik, J. Grytczuk, and Z. Miechowicz, Generalized arboricity of graphs with large girth. *Discrete Math.* **342** (2019), no. 5, 1343–1350 Zbl 1407.05085 MR 3907838

[9] J. Beck, An algorithmic approach to the Lovász local lemma. I. *Random Structures Algorithms* **2** (1991), no. 4, 343–365

[10] A. Bernshteyn, The local cut lemma. *European J. Combin.* **63** (2017), 95–114 Zbl 1365.05288 MR 3645787

[11] R. Bissacot, R. Fernández, and A. Procacci, On the convergence of cluster expansions for polymer gases. *J. Stat. Phys.* **139** (2010), no. 4, 598–617 Zbl 1196.82135 MR 2638929

[12] R. Bissacot, R. Fernández, A. Procacci, and B. Scoppola, An improvement of the Lovász local lemma via cluster expansion. *Combin. Probab. Comput.* **20** (2011), no. 5, 709–719 Zbl 1233.05196 MR 2825585

[13] J. Böttcher, Y. Kohayakawa, and A. Procacci, Properly coloured copies and rainbow copies of large graphs with small maximum degree. *Random Structures Algorithms* **40** (2012), no. 4, 425–436 Zbl 1244.05087 MR 2925305

[14] J. Cai, X. Li, and G. Yan, Improved upper bound for the degenerate and star chromatic numbers of graphs. *J. Comb. Optim.* **34** (2017), no. 2, 441–452 Zbl 1376.05050 MR 3672402

[15] S. Camungol and N. Rampersad, Avoiding approximate repetitions with respect to the longest common subsequence distance. *Involve* **9** (2016), no. 4, 657–666 Zbl 1362.68238 MR 3530205

[16] R. L. Dobrushin, Perturbation methods of the theory of Gibbsian fields. In *Lectures on probability theory and statistics (Saint-Flour, 1994)*, pp. 1–66, Lecture Notes in Math. 1648, Springer, Berlin, 1996 Zbl 0871.60086 MR 1600880

[17] M. Drmota, Combinatorics and asymptotics on trees. *Cubo* **6** (2004), no. 2, 105–136 Zbl 1080.05004 MR 2092045

[18] V. Dujmović, G. Joret, J. Kozik, and D. R. Wood, Nonrepetitive colouring via entropy compression. *Combinatorica* **36** (2016), no. 6, 661–686 Zbl 1399.05071 MR 3597586

[19] P. Erdős and L. Lovász, Problems and results on 3-chromatic hypergraphs and some related questions. In *Infinite and finite sets (Keszthely, 1973),* Vol. II, pp. 609–627, Colloq. Math. Soc. János Bolyai 10, North-Holland, Amsterdam, 1975 Zbl 0315.05117 MR 0382050

[20] L. Esperet and A. Parreau, Acyclic edge-coloring using entropy compression. *European J. Combin.* **34** (2013), no. 6, 1019–1027 Zbl 1285.05056 MR 3037985

[21] R. Fernández and A. Procacci, Cluster expansion for abstract polymer models. New bounds from an old approach. *Comm. Math. Phys.* **274** (2007), no. 1, 123–140 Zbl 1206.82148 MR 2318850

[22] P. M. S. Fialho, Abstract polymer gas: a simple inductive proof of the Fernández–Procacci criterion. *J. Stat. Phys.* **178** (2020), no. 6, 1354–1361 Zbl 1436.82003 MR 4081232

[23] P. M. S. Fialho, B. N. B. de Lima, and A. Procacci, A new bound on the acyclic edge chromatic number. *Discrete Math.* **343** (2020), no. 11, article id. 112037 Zbl 1447.05078 MR 4118384

[24] P. Flajolet and R. Sedgewick, *Analytic combinatorics*. Cambridge University Press, Cambridge, 2009 Zbl 1165.05001 MR 2483235

[25] N. Francetić and B. Stevens, Asymptotic size of covering arrays: an application of entropy compression. *J. Combin. Des.* **25** (2017), no. 6, 243–257 Zbl 1370.05033 MR 3646004

[26] I. Giotis, L. Kirousis, K. I. Psaromiligkos, and D. M. Thilikos, Acyclic edge coloring through the Lovász local lemma. *Theoret. Comput. Sci.* **665** (2017), 40–50 Zbl 1357.05042 MR 3603327

[27] D. Gonçalves, M. Montassier, and A. Pinlou, Entropy compression method applied to graph colorings. 2014, arXiv:1406.4380

[28] C. Gruber and H. Kunz, General properties of polymer systems. *Comm. Math. Phys.* **22** (1971), 133–161 MR 321473

[29] J. Grytczuk, J. Kozik, and P. Micek, New approach to nonrepetitive sequences. *Random Structures Algorithms* **42** (2013), no. 2, 214–225 Zbl 1349.68131 MR 3019398

[30] D. G. Harris, Oblivious resampling oracles and parallel algorithms for the lopsided Lovász local lemma. In *Proceedings of the Thirtieth Annual ACM–SIAM Symposium on Discrete Algorithms*, pp. 841–860, SIAM, Philadelphia, P.A., 2019 Zbl 1431.68128 MR 3909521

[31] D. G. Harris and A. Srinivasan, A constructive Lovász local lemma for permutations. *Theory Comput.* **13** (2017), article id. 17 Zbl 1387.68293 MR 3743434

[32] N. J. A. Harvey and J. Vondrák, An algorithmic proof of the Lovász local lemma via resampling oracles. *SIAM J. Comput.* **49** (2020), no. 2, 394–428 Zbl 1433.68603 MR 4082011

[33] K. Kolipaka and M. Szegedy, Moser and Tardos meet Lovász. In *STOC'11 – Proceedings of the 43rd ACM Symposium on Theory of Computing*, pp. 235–243, ACM, New York, 2011 Zbl 1288.68129 MR 2931973

[34] V. Kolmogorov, Commutativity in the algorithmic Lovász local lemma. *SIAM J. Comput.* **47** (2018), no. 6, 2029–2056 Zbl 1429.60018 MR 3874020

[35] R. Kotecký and D. Preiss, Cluster expansion for abstract polymer models. *Comm. Math. Phys.* **103** (1986), no. 3, 491–498 Zbl 0593.05006 MR 832923

[36] J. Kozik and D. Shabanov, Improved algorithms for colorings of simple hypergraphs and applications. *J. Combin. Theory Ser. B* **116** (2016), 312–332 Zbl 1327.05113 MR 3425248

[37] L. Kirousis and L. Livieratos, A simple algorithmic proof of the symmetric lopsided Lovász local lemma. In *Learning and Intelligent Optimization – Lion 2018,* pp. 49–63, Lecture Notes in Comput. Sci. 11353, Springer, Cham, 2019,

[38] R. A. Moser and G. Tardos, A constructive proof of the general Lovász local lemma. *J. ACM* **57** (2010), no. 2, Article id. 11 Zbl 1300.60024 MR 2606086

[39] S. Ndreca, A. Procacci, and B. Scoppola, Improved bounds on coloring of graphs. *European J. Combin.* **33** (2012), no. 4, 592–609 Zbl 1244.05094 MR 2864444

[40] P. Ochem and A. Pinlou, Application of entropy compression in pattern avoidance. *Electron. J. Combin.* **21** (2014), no. 2, article id. 2.7 Zbl 1299.68046 MR 3210641

[41] W. Pegden, An extension of the Moser–Tardos algorithmic local lemma. *SIAM J. Discrete Math.* **28** (2014), no. 2, 911–917 Zbl 1301.60013 MR 3215059

[42] A. Procacci and R. Sanchis, Perfect and separating hash families: new bounds via the algorithmic cluster expansion local lemma. *Ann. Inst. Henri Poincaré D* **5** (2018), no. 2, 153–171 Zbl 1391.05253 MR 3813213

[43] J. Przybyło, On the facial Thue choice index via entropy compression. *J. Graph Theory* **77** (2014), no. 3, 180–189 Zbl 1303.05067 MR 3258717

[44] J. Przybyło, J. Schreyer, and E. Škrabul'áková, On the facial Thue choice number of plane graphs via entropy compression method. *Graphs Combin.* **32** (2016), no. 3, 1137–1153 Zbl 1338.05057 MR 3489727

[45] A. D. Scott and A. D. Sokal, The repulsive lattice gas, the independent-set polynomial, and the Lovász local lemma. *J. Stat. Phys.* **118** (2005), no. 5-6, 1151–1261 Zbl 1107.82013 MR 2130890

[46] J.-S. Sereni and J. Volec, A note on acyclic vertex-colorings. *J. Comb.* **7** (2016), no. 4, 725–737 Zbl 1350.05040 MR 3538161

[47] J. B. Shearer, On a problem of Spencer. *Combinatorica* **5** (1985), no. 3, 241–245 Zbl 0587.60012 MR 837067

[48] Y. G. Sinaĭ, *Theory of phase transitions: rigorous results*. International Series in Natural Philosophy 108, Pergamon Press, Oxford and Elmsford, N.Y., 1982 Zbl 0537.60097 MR 691854

[49] T. Tao, Moser's entropy compression argument. What's new blog, 2009 https://terrytao.wordpress.com/2009/08/05/mosers-entropy-compression-argument/

Communicated by Adrian Tanasă

Received 4 January 2020; revised 8 July 2020.

Paula M. S. Fialho
Campus Pampulha, Departamento de Matemática – Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, Caixa Postal 702, Belo Horizonte – Minas Gerais CEP 31270-901, Brazil; paulamendes26@hotmail.com

Bernardo N. B. de Lima
Campus Pampulha, Departamento de Matemática – Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, Caixa Postal 702, Belo Horizonte – Minas Gerais CEP 31270-901, Brazil; bnblima@mat.ufmg.br

Aldo Procacci
Campus Pampulha, Departamento de Matemática – Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, Caixa Postal 702, Belo Horizonte – Minas Gerais CEP 31270-901, Brazil; aldo@ufmg.br