# Commutativity in the Algorithmic Lovász Local Lemma

Vladimir Kolmogorov
IST Austria
Klosterneuburg, Austria
vnk@ist.ac.at

***Abstract*—We consider the recent formulation of the Algorithmic Lovász Local Lemma [1], [2] for finding objects that avoid "bad features", or "flaws". It extends the Moser-Tardos resampling algorithm [3] to more general discrete spaces. At each step the method picks a flaw present in the current state and "resamples" it using a "resampling oracle" provided by the user. However, it is less flexible than the Moser-Tardos method since [1], [2] require a specific flaw selection rule, whereas [3] allows an arbitrary rule (and thus can potentially be implemented more efficiently).**

**We formulate a new "commutativity" condition, and prove that it is sufficient for an arbitrary rule to work. It also enables an efficient parallelization under an additional assumption. We then show that existing resampling oracles for perfect matchings and permutations do satisfy this condition.**

**Finally, we generalize the precondition in [2] (in the case of symmetric potential causality graphs). This unifies special cases that previously were treated separately.**

***Index Terms*—component; formatting; style; styling;**

## I. Introduction

Let $\Omega$ be a (large) set of objects and $F$ be a set of *flaws*, where a flaw $f \in F$ is some non-empty set of "bad" objects, i.e. $f \subseteq \Omega$. Flaw $f$ is said to be *present in* $\sigma$ if $\sigma \in f$. Let $F_\sigma = \{f \in F \,|\, \sigma \in f\}$ be the set of flaws present in $\sigma$. Object $\sigma$ is called *flawless* if $F_\sigma = \varnothing$.

The existence of flawless objects can often be shown via a probabilistic method. First, a probability measure $\omega$ on $\Omega$ is introduced, then flaws in $F$ become (bad) events that should be avoided. Proving the existence of a flawless object is now equivalent to showing that the probability of avoiding all bad events is positive. This holds if, for example, all events $f \in F$ are independent and the probability of each $f$ is smaller than 1. The well-known Lovász Local Lemma (LLL) [4] is a powerful tool that can handle a (limited) dependency between the events. Roughly speaking, it states that if the dependency graph is sparse enough (e.g. has a bounded degree) and the probabilities of individual bad events are sufficiently small then a flawless object is guaranteed to exist.

LLL has been the subject of intensive research, see e.g. [5] for a relatively recent survey. One of the milestone results was the *constructive* version of LLL by Moser and Tardos [3]. It applies to the *variable model* in which $\Omega = \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$ for some discrete sets $\mathcal{X}_i$, event $f$ depends on a small subset of variables denoted as $\mathtt{vbl}(f) \subseteq [n]$, and two events $f, g$ are declared to be dependent if $\mathtt{vbl}(f) \cap \mathtt{vbl}(g) \neq \varnothing$. The algorithm proposed in [3] is strikingly simple: (i) sample each variable $\sigma_i$ for $i \in [n]$ according to its distribution; (ii) while $F_\sigma$ is non-empty, pick an arbitrary flaw $f \in F_\sigma$ and resample all variables $\sigma_i$ for $i \in \mathtt{vbl}(f)$. Moser and Tardos proved that if the LLL condition in [4] is satisfied then the expected number of resampling is small (polynomial for most of the known applications).

The recent development has been extending algorithmic LLL beyond the variable model, and in particular to non-Cartesian spaces. The first such work was by Harris and Srinivasan [6], who considered the space of permutations. Achlioptas and Iliopoulos [7] introduced a more abstract framework where the behaviour of the algorithm is specified by a certain multigraph. Harvey and Vondrák proposed a framework with *regenerating resampling oracles* [1], providing a more direct connection to LLL. Achlioptas and Iliopoulos [2] extended the framework to more general resampling oracles.

In this paper we study this setting from [1], [2]. It does not assume any particular structure on sets $\Omega$ and $F$. Instead, for each object $\sigma \in \Omega$ and flaw $f \in F_\sigma$ the user must provide a "resampling oracle" specified by a set of *actions* $A(f, \sigma) \subseteq \Omega$ that can be taken to "address" flaw $f$, and a probability distribution $\rho(\sigma'|f, \sigma)$ over $\sigma' \in A(f, \sigma)$. At each step the algorithm selects a certain flaw $f \in F_\sigma$, samples an action $\sigma' \in A(f, \sigma)$ according to $\rho(\sigma'|f, \sigma)$, and goes there. This framework captures the Moser-Tardos algorithm [3], and can also handle other scenarios such as permutations and perfect matchings (in which case $\Omega$ cannot be expressed as a Cartesian product).

One intriguing difference between the methods of [3] and [7], [1], [2] is that [3] allows an arbitrary rule for selecting a flaw $f \in F_\sigma$, whereas [7], [1], [2] require a specific rule (which depends on a permutation $\pi$ of $F$ chosen in advance)[1]. We will say that a resampling algorithm is *flexible* if it is guaranteed to work with any flaw selection rule. We argue that flexibility can lead to a much more efficient practical implementation: it is not necessary to examine all flaws in $F_\sigma$, the first found flaw will suffice. If the list of current flaws is updated dynamically then flexibility could potentially eliminate the need for a costly data structure (such as a priority queue) and thus save a factor of $\Theta(\log n)$ in the complexity.

[1]The papers [7], [2] actually allowed more freedom in the choice of permutation $\pi$, e.g. it may depend on the iteration number. However, once $\pi$ has been chosen, the algorithm should still examine some "current" set of flaws and pick the lowest one with respect to $\pi$.

0272-5428/16 $31.00 © 2016 IEEE
DOI 10.1109/FOCS.2016.88

The rule may also affect the number of resamplings in practice; experimentally, the selection process matters, as noted in [5].

Achlioptas and Iliopoulos discuss flaw selection rules in [7, Section 4.3], and remark that they do not see how to accommodate arbitrary rules in their framework. It is known, however, that in special cases flexible rules can be used even beyond the variable model. Namely, through a lengthy and a complicated analysis Harris and Srinivasan [6] managed to show the correctness of a resampling algorithm for permutations, and did not make assumptions on the flaw selection rule in their proof. They also proved a better bound for the parallel version of the algorithm.

This paper aims to understand which properties of the problem enable flexibility and parallelism. Our contributions are as follows.

- We formulate a new condition that we call "commutativity", and prove that it is sufficient for flexibility.
- We prove that it gives a better bound on the number of rounds of the parallel version of the algorithm. In particular, we show how to use commutativity for handling "partial execution logs" instead of "full execution logs" (which is required for analyzing the parallel version).
- We show that existing resampling oracles for permutations [6] and perfect matchings in complete graphs [1] are commutative. (In fact, we treat both cases in a single framework). Thus, we provide a simpler proof of the result in [6] and generalize it to other settings, in particular to perfect matchings in certain graphs (for which existing algorithms require specific rules).
- We generalize the condition in [2] for the algorithmic LLL to work (in the case of symmetric potential causality graphs). The new condition unifies special cases that were treated separately in [2].

To our knowledge, our commutativity condition captures all previously known cases when the flaw selection rules was allowed to be arbitrary.

**Other related work** Applications that involve non-Cartesian spaces $\Omega$ (such as permutations, matchings and spanning trees) have often been tackled via the *Lopsided LLL* [8]; we refer to [9], [10] for a comprehensive survey. On the level of techniques there is some connection between this paper and a recent work by Knuth [11]; we discuss this in Section III.

## II. Background and Preliminaries

First, we give a formal description of the algorithm from [1], [2]. Assume that for each object $\sigma \in \Omega$ and each flaw $f \in F_\sigma$ there is a non-empty set of *actions* $A(f, \sigma) \subseteq \Omega$ that can be taken for "addressing" flaw $f$ at $\sigma$, and a probability distribution $\rho(\sigma'|f, \sigma)$ over $\sigma' \in A(f, \sigma)$. Note, by definition $A(f, \sigma)$ is the support of distribution $\rho(\cdot|f, \sigma)$. The collection of all resampling oracles will be denoted as $\rho$. We fix some probability distribution $\omega$ on $\Omega$ with $\omega(\sigma) > 0$ for all $\sigma \in \Omega$ (it will be used later for formulating various conditions). Note that our notation is quite different from that of Harvey and Vondrák [1].[2] The algorithm can now be stated as follows.

**Algorithm 1** Random walk. Input: initial distribution $\omega^{\mathtt{init}}$ over $\Omega$, strategy $\Lambda$.

1: sample $\sigma \in \Omega$ according to $\omega^{\mathtt{init}}$
2: **while** $F_\sigma$ non-empty **do**
3: select flaw $f \in F_\sigma$ according to $\Lambda$
4: sample $\sigma' \in A(f, \sigma)$ according to distribution $\rho(\sigma'|f, \sigma)$, set $\sigma \leftarrow \sigma'$.
5: **end while**

Clearly, if the algorithm terminates then it produces a flawless object $\sigma$. The works [7], [1], [2] used specific strategies $\Lambda$. As stated in the introduction, our goal is to understand when an arbitrary strategy can be used. This means that flaw $f$ in line 3 is selected according to some distribution which is a function of the entire past execution history[3]. Note that if flaw $f \in F_\sigma$ in line 3 depends only on $\sigma$ then the algorithm can be viewed a random walk in a Markov chain with states $\Omega$, while in a more general case the walk can be non-Markovian.

### A. Walks and the potential causality graph

We say that $\sigma \xrightarrow{f} \sigma'$ is a (valid) walk if it is possible to get from state $\sigma$ to $\sigma'$ by "addressing" flaw $f$ as described in the algorithm, i.e. if two conditions hold: $f \in F_\sigma$ and $\sigma' \in A(f, \sigma)$. Whenever we write $\sigma \xrightarrow{f} \sigma'$, we mean that it is a valid walk.

In many applications resampling oracles satisfy a special condition called *atomicity* [7].

**Definition 1.** *$\rho$ is called* atomic *if for any $f \in F$ and $\sigma' \in \Omega$ there exists at most one object $\sigma \in \Omega$ such that $\sigma \xrightarrow{f} \sigma'$.*

Next, we need to describe "dependences" between flaws in $F$. Let $\sim$ be some symmetric relation on $F$ (so that $(F, \sim)$ is an undirected graph). It is assumed to be fixed throughout the paper. For a flaw $f \in F$ let $\Gamma(f) = \{g \in F \mid f \sim g\}$ be the set of neighbors of $f$. Note, we may or may not have $f \sim f$, and so $\Gamma(f)$ may or may not contain $f$. We will denote $\Gamma^+(f) = \Gamma(f) \cup \{f\}$, and also $\Gamma(S) = \bigcup_{f \in S} \Gamma(f)$ and $\Gamma^+(S) = \bigcup_{f \in S} \Gamma^+(f)$ for a subset $S \subseteq F$.

**Definition 2.** *Undirected graph $(F, \sim)$ is called a* potential causality graph *for $\rho$ if for any walk $\sigma \xrightarrow{f} \sigma'$ there holds $F_{\sigma'} \subseteq (F_\sigma - \{f\}) \cup \Gamma(f)$.*

In other words, $\Gamma(f)$ must contain all flaws that can appear after addressing flaw $f$ at some state. Also, $\Gamma(f)$ must contain $f$ if addressing $f$ at some state can fail to eradicate $f$.

[2]"Flaws" $f$ correspond to "bad events" $E_i$ in [1]. The distribution over $\Omega$ was denoted in [1] as $\mu$, the states of $\Omega$ as $\omega$, and the resampling oracle for the bad event $E_i$ at state $\omega \in \Omega$ as $r_i(\omega)$.

[3]The description of the algorithm in [3] says "*pick an arbitrary violated event*". This is consistent with our definition of an "arbitrary strategy": in the analysis Moser and Tardos mention that this selection must come from some fixed procedure (either deterministic or randomized), so that expected values are well-defined.

Note that in Definition 2 we deviated slightly from [7], [2]: in their analysis the potential causality graph was *directed* and therefore in certain cases could capture more information about $D$. While directed graphs do matter in some applications (see examples in [7], [2]), we believe that in a typical application the potential causality relation is symmetric. Using an undirected graph will be essential for incorporating commutativity.

A subset $S \subseteq F$ will be called *independent* if for any *distinct* $f, g \in S$ we have $f \nsim g$. (Thus, loops $f \sim f$ in the graph $(F, \sim)$ do not affect the definition of independence). For a subset $S \subseteq F$ we denote $\mathtt{Ind}(S) = \{T \subseteq S \mid T \text{ is independent}\}$.

### B. Commutativity

We now formulate new conditions that will allow an arbitrary flaw selection rule to be used.

**Definition 3.** $(\rho, \sim)$ *is called* weakly commutative *if there exists a mapping* SWAP *that sends any walk* $\sigma_1 \xrightarrow{f} \sigma_2 \xrightarrow{g} \sigma_3$ *with* $f \nsim g$ *to another valid walk* $\sigma_1 \xrightarrow{g} \sigma_2' \xrightarrow{f} \sigma_3$*, and this mapping is injective.*

Note that in the atomic case the definition can be simplified. Namely, $(\rho, \sim)$ is weakly commutative if and only if it satisfies the following condition:

- *For any walk* $\sigma_1 \xrightarrow{f} \sigma_2 \xrightarrow{g} \sigma_3$ *with* $f \nsim g$ *there exists state* $\sigma_2' \in \Omega$ *such that* $\sigma_1 \xrightarrow{g} \sigma_2' \xrightarrow{f} \sigma_3$ *is also a walk.*

Indeed, by atomicity the state $\sigma_2'$ is unique, and so mapping SWAP in Definition 3 is constructed in a natural way. This mapping is reversible and thus injective.

For several results we will also need a stronger property.

**Definition 4.** $(\rho, \sim)$ *is called* strongly commutative *(or just* commutative*) if for any walk* $\tau = \sigma_1 \xrightarrow{f} \sigma_2 \xrightarrow{g} \sigma_3$ *with* $f \nsim g$ *and* $\mathtt{SWAP}(\tau) = \sigma_1 \xrightarrow{g} \sigma_2' \xrightarrow{f} \sigma_3$ *there holds*

$$\rho(\sigma_2|f, \sigma_1)\rho(\sigma_3|g, \sigma_2) = \rho(\sigma_2'|g, \sigma_1)\rho(\sigma_3|f, \sigma_2') \qquad (1)$$

It is straightforward to check that strong commutativity holds in the variable model of Moser and Tardos. In fact, an additional property holds: for any $\sigma_1 \xrightarrow{f} \sigma_2 \xrightarrow{g} \sigma_3$ with $f \nsim g$ there exists exactly one state $\sigma_2' \in \Omega$ such that $\sigma_1 \xrightarrow{g} \sigma_2' \xrightarrow{f} \sigma_3$. Checking strong commutativity for non-Cartesian spaces $\Omega$ is more involved; we refer to the full version of the paper [12] for details.

### C. Parallel version

We will also consider the following version of the algorithm (see Algorithm 2). It is equivalent to the parallel algorithm of Moser and Tardos [3] in the case of the variable model, and to the parallel algorithm of Harris and Srinivasan [6] in the case of permutations. It is also closely related to the "MaximalSetResample" algorithm of Harvey and Vondrák [1] (see below).

Lines 3-8 will be called a *round*. In some cases each round admits an efficient parallel implementation (with a polylogarithmic running time). For example, this holds in

**Algorithm 2** Parallel random walk.

1: sampe $\sigma \in \Omega$ according to distribution $\omega^{\mathtt{init}}$
2: **while** $F_\sigma$ non-empty **do**
3: set $I = \varnothing$
4: **while** set $F_\sigma - \Gamma^+(I)$ is non-empty **do**
5: pick some $f \in F_\sigma - \Gamma^+(I)$
6: sample $\sigma' \in A(f, \sigma)$ according to $\rho(\sigma'|f, \sigma)$, set $\sigma \leftarrow \sigma'$.
7: set $I \leftarrow I \cup \{f\}$
8: **end while**
9: **end while**

the variable model of Moser and Tardos [3]. Also, Harris and Srinivasan [6] presented an efficient implementation for permutations. Accordingly, we will be interested in the number of rounds of the algorithm.

Note, during round $r$ set $F_\sigma - \Gamma^+(I)$ in line 5 shrinks from iteration to iteration (and so flaw $f$ in line 5 satisfies $f \in F_{\sigma_r}$, where $\sigma_r$ is the state in the beginning of round $r$). This property can be easily verified using induction and Definition 2.

**$\pi$-stable strategy** Let us fix a total order $\preceq_\pi$ on $F$ defined by some permutation $\pi$ of $F$. Consider a version of Algorithm 2 where flaw $f$ in line 5 is selected as the lowest flaw in $F_\sigma - \Gamma^+(I)$ (with respect to $\preceq_\pi$). This corresponds to Algorithm 1 with a specific strategy $\Lambda$; this strategy will be called $\pi$*-stable*. It coincides with the MaximalSetResample algorithm of Harvey and Vondrák [1].

Although we focus on the commutative case, we will also state results for $\pi$-stable strategies since they follow automatically from the proof (which is based on the analysis of $\pi$*-stable walks*).

### D. Algorithmic LLL conditions

In this section we formulate sufficient conditions under which a flawless object will be guaranteed to exist. The conditions involve two vectors, $\lambda$ and $\mu$. Roughly speaking, $\lambda$ characterizes resampling oracles and $\mu$ characterizes graph $(F, \sim)$.

**Definition 5.** *The pair* $(\rho, \sim)$ *is said to satisfy Algorithmic LLL conditions if there exist vectors* $\lambda, \mu \in \mathbb{R}^{|F|}$ *such that*

$$\lambda_f \geq \sum_{\sigma \in f : \sigma' \in A(f,\sigma)} \rho(\sigma'|f, \sigma)\frac{\omega(\sigma)}{\omega(\sigma')} \quad \forall f \in F, \sigma' \in \Omega \qquad (2a)$$

$$\frac{\lambda_f}{\mu_f} \sum_{S \in \mathtt{Ind}(\Gamma(f))} \mu(S) \leq \theta \qquad \forall f \in F \qquad (2b)$$

*where* $\theta \in (0, 1)$ *is some constant and* $\mu(S) = \prod_{g \in S} \mu_g$.

Of course, vector $\lambda$ can be easily eliminated from (2). However, it is convenient to have it explicitly since in many cases it has a natural interpretation. Achlioptas and Iliopoulos [2] called $\lambda$ *flaw charges*, though instead of (2a) they used slightly stronger conditions. Namely, they considered the following cases; it is straightforward to check that in each one of them vector $\lambda$ satisfies (2a):

- The case from their earlier work [7], which in the current terminology can be described as follows: $\omega$ is a uniform distribution over $\Omega$, $\rho(\cdot|f,\sigma)$ is a uniform distribution over $A(f,\sigma)$, and $\rho$ is atomic. They then defined $\lambda_f = 1/\min_{\sigma\in f}|A(f,\sigma)|$.
- Regenerating resampling oracles of Harvey and Vondrák [1] specified by the equation

$$\frac{1}{\omega(f)}\sum_{\sigma\in f}\rho(\sigma'|f,\sigma)\omega(\sigma)=\omega(\sigma') \qquad \forall f\in F,\sigma'\in\Omega$$

  where $\omega(f)=\sum_{\sigma\in f}\omega(\sigma)$. In this case Achlioptas and Iliopoulos [2] defined $\lambda_f=\omega(f)$.
- In the general case, [2] defined flaw charges via

$$\lambda_f = b_f \max_{\sigma\in f,\sigma'\in A(f,\sigma)}\left\{\rho(\sigma'|f,\sigma)\frac{\omega(\sigma)}{\omega(\sigma')}\right\}$$

  where $b_f=\max_{\sigma'\in\Omega}|\{\sigma\in f:\sigma'\in A(f,\sigma)\}|$.

**Remark 1.** *An alternative condition that appeared in the literature (for certain $\lambda$'s) is*

$$\frac{\lambda_f}{\mu_f}\sum_{S\subseteq\Gamma(f)}\mu(S)=\frac{\lambda_f}{\mu_f}\prod_{g\in\Gamma(f)}(1+\mu_g)\le\theta \tag{3}$$

*Clearly,* (2b) *is weaker than* (3)*. We mention that* (3) *is analogous to the original LLL condition in [4], while* (2b) *corresponds to the* cluster expansion *improvement by Bissacot et al. [13] (with the matching algorithmic version by Pedgen [14] who considered the variable model of Moser and Tardos). It is known that the cluster expansion version can give better results for some applications, see e.g. [15], [16], [1].*

**Shearer's condition** Shearer [17] gave a sufficient and necessary condition for a general LLL to hold for a given dependency graph. Kolipaka and Szegedy [18] showed that this condition is sufficient for the Moser-Tardos algorithm, while Harvey and Vondrák [1] generalized the analysis to regenerating resampling oracles. We will show that the same analysis holds for the framework considered in this paper.

Consider vector $p\in\mathbb{R}^{|F|}$. For a subset $S\subseteq F$ denote $p^S=\prod_{f\in S}p_f$; this is a monomial in variables $\{p_f\mid f\in F\}$. Also, define polynomial $q_S$ as follows:

$$q_S=q_S(p)=\sum_{I:S\subseteq I\in\mathtt{Ind}(F)}(-1)^{|I|-|S|}p^I \tag{4}$$

**Definition 6.** *Vector $p$ is said to satisfy the Shearer's condition if $q_S(p)\ge 0$ for all $S\subseteq F$, and $q_\varnothing(p)>0$.*

*The pair $(\rho,\sim)$ is said to satisfy Shearer's condition if there exist vector $p$ satisfying Shearer's condition, vector $\lambda$ satisfying* (2a)*, and a constant $\theta\in(0,1)$ such that $\lambda_f\le\theta\cdot p_f$ for all $f\in F$.*

## III. Our results

First, we state our results for the sequential version (Algorithm 1). Unless mentioned otherwise, the flaw selection strategy and the initial distribution $\omega^{\mathtt{init}}$ are assumed to be arbitrary.

**Theorem 7.** *Suppose that $(\rho,\sim)$ satisfies either condition* (2) *or the Shearer's condition, and one of the following holds:*

(a) *Algorithm 1 uses a $\pi$-stable strategy.*
(b) *$(\rho,\sim)$ is weakly commutative and atomic.*
(c) *$(\rho,\sim)$ is strongly commutative.*

*Define*

$$\gamma^{\mathtt{init}} = \max_{\sigma\in\Omega}\frac{\omega^{\mathtt{init}}(\sigma)}{\omega(\sigma)},$$

$$\mathtt{Ind}^{\mathtt{init}} = \begin{cases}\bigcup_{\sigma\in\mathtt{supp}(\omega^{\mathtt{init}})}\mathtt{Ind}(F_\sigma) & \textit{in cases (a,b)}\\ \mathtt{Ind}(F) & \textit{in the case (c)}\end{cases}$$

*where $\mathtt{supp}(\omega^{\mathtt{init}})=\{\sigma\in\Omega\mid\omega^{\mathtt{init}}(\sigma)>0\}$ is the support of $\omega^{\mathtt{init}}$. The probability that Algorithm 1 produces a flawless object in fewer than $T+r$ steps is at least $1-\theta^r$ where*

$$T = \frac{1}{\log\theta^{-1}}\left(\log\gamma^{\mathtt{init}}+\log\sum_{R\in\mathtt{Ind}^{\mathtt{init}}}\mu(R)\right) \tag{5}$$

*and $\mu(R)=\prod_{f\in R}\mu_f$ (in the case of condition* (2)*) or $\mu(R)=\frac{q_R(p)}{q_\varnothing(p)}$ (in the case of the Shearer's condition).*

Note that part (a) of Theorem 7 is a minor variation of existing results [1], [2] (except that our precondition (2a) unifies conditions in previous works - see Section II-D):

- Harvey and Vondrák [1] proved Theorem 7(a) in the case of regenerating oracles and distributions $\omega^{\mathtt{init}}=\omega$, with a slightly different expression for $T$.
- Achlioptas and Iliopoulos [2] proved the result for the "`RecursiveWalk`" strategy in the special cases described in Section II-D (and assuming condition (2b)).

Parts (b,c) are new results.

**Remark 2.** *The possibility of using distribution $\omega^{\mathtt{init}}$ which is different from $\omega$ was first proposed by Achlioptas and Iliopoulos in [7]. Namely, they used a distribution with $|\mathtt{supp}(\omega^{\mathtt{init}})|=1$, and later extended it to arbitrary distributions $\omega^{\mathtt{init}}$ in [2]. There is a trade-off in choosing $\omega^{\mathtt{init}}$: smaller $\mathtt{supp}(\omega^{\mathtt{init}})$ leads to a smaller set $\mathtt{Ind}^{\mathtt{init}}$ but increases the constant $\gamma^{\mathtt{init}}$. It is argued in [2] that using $\omega^{\mathtt{init}}\ne\omega$ can be beneficial when sampling from $\omega$ is a difficult problem, or when the number of flaws is exponentially large.*

Next, we analyze the parallel version.

**Theorem 8.** *Suppose that $(\rho,\sim)$ satisfies either condition* (2) *or the Shearer's condition, and is strongly commutative. Then the probability that Algorithm 2 produces a flawless object in* *fewer than $T + r$ rounds is at least $1 - \theta^r$ where*

$$T \quad = \quad \frac{1}{\log \theta^{-1}} \left( \log \gamma^{\mathtt{init}} + \log \sum_{f \in F} \mu_f \right) \qquad (6)$$

*where $\gamma^{\mathtt{init}}$ is the constant from Theorem 7, and $\mu_f = \frac{q_{\{f\}}(p)}{q_{\varnothing}(p)}$ (in the case of the Shearer's condition).*

**Our techniques** The general idea of the proofs is to construct a "swapping mapping" that transforms "walks" (which are possible executions of the algorithm) to some canonical form by applying swap operations from Definition 3. Importantly, we need to make sure that the mapping is injective: this will guarantee that the sum over original walks is smaller or equal than the sum over "canonical walks". We then upper-bound the latter sum using some standard techniques [18], [1]. We use two approaches:

1) Theorem 7(b): transforming walks to "forward stable sequences" (a *forward-looking analysis*). This works only in the atomic case (under the weak commutativity assumption), and can make use of the knowledge of the set $\mathtt{supp}(\omega^{\mathtt{init}})$, leading to a tighter definition of the set $\mathtt{Ind}^{\mathtt{init}}$.
2) Theorems 7(c) and 8: transforming walks to "backward stable sequences" (a *backward-looking analysis*). This works in the non-atomic cases, but requires strong commutativity. In this approach the "roots" of stable sequences are on the right, and have no connection to $\omega^{\mathtt{init}}$; this means that we must use $\mathtt{Ind}^{\mathtt{init}} = \mathtt{Ind}(F)$. Analyzing the parallel version requires dealing with "partial execution logs" instead of "full execution logs". It appears that this is possible only with backward sequences.

Note that previously a backward-looking analysis (with either "stable sequences" or "witness trees") was used for the variable model of Moser and Tardos [3], [18], [14], while a forward-looking analysis was used for LLL versions on non-Cartesian spaces [7], [1], [2] and also on Cartesian spaces [19].

After the first version of this work [20] we learned about a recent book draft by Knuth [11]. He considers the variable model of Moser-Tardos, and gives an alternative proof of algorithm's correctness which is also based on swapping arguments (justified by a technique of "coupling" two random sources, similar to [3]). We emphasize that we go beyond the variable model, in which case justifying "swapping" seems to require different techniques.

The next section gives a sketch of the proofs of Theorems 7 and 8; complete proofs can be found in the full version of this paper [12]. For convenience, we use the same numeration of statements in both versions. The most technical part is probably constructing an injective swapping mapping for transforming to backward stable sequences. In [12] we also describe our third result, which is a proof of strong commutativity of some existing resampling oracles. Finally, we consider one application, namely rainbow matchings in complete graphs.

## IV. PROOF SKETCH

We write $f \cong g$ for flaws $f, g \in F$ if either $f \sim g$ or $f = g$ (and $f \ncong g$ otherwise).

A *walk* of length $t$ is a sequence $\tau = \sigma_1 \xrightarrow{w_1} \sigma_2 \ldots \sigma_t \xrightarrow{w_t} \sigma_{t+1}$ such that $w_i \in F_{\sigma_i}$ and $\sigma_{i+1} \in A(w_i, \sigma_i)$ for $i \in [t]$. Its length is denoted as $|\tau| = t$. For such a walk we define quantity

$$p(\tau) = \omega^{\mathtt{init}}(\sigma_1) \cdot \prod_{i \in [t]} \rho(\sigma_{i+1} | w_i, \sigma_i) \qquad (7)$$

Let $\Lambda$ be the strategy for selecting flaws used in Algorithm 1. We assume that this strategy is deterministic, i.e. the flaw $w_i$ in a walk $\tau = \sigma_1 \xrightarrow{w_1} \ldots \xrightarrow{w_{i-1}} \sigma_i \xrightarrow{w_i} \ldots$ is uniquely determined by the previous history $\tau_i = \sigma_1 \xrightarrow{w_1} \ldots \xrightarrow{w_{i-1}} \sigma_i$. This assumption can be made w.l.o.g. (see [12]).

A walk $\tau$ of length $t$ that can be produced by Algorithm 1 with a positive probability will be called a *bad $t$-trajectory*. Equivalently, it is a walk that starts at a state $\sigma \in \mathtt{supp}(\omega^{\mathtt{init}})$ and follows strategy $\Lambda$. Note that it goes only through flawed states (except possibly the last state). Let $\mathtt{Bad}(t)$ be the set of all bad $t$-trajectories. Clearly, for any $\tau \in \mathtt{Bad}(t)$ the probability that the algorithm will produce $\tau$ equals $p(\tau)$, as defined in (7). This gives

**Proposition 9.** *The probability that Algorithm 1 takes $t$ steps or more equals $\sum_{\tau \in \mathtt{Bad}(t)} p(\tau)$.*

If $W = w_1 \ldots w_t$ is the complete sequence of flaws in a walk $\tau$ then we will write $\tau \doteq W$. If we want to indicate certain intermediate states of $\tau$ then we will write them in square brackets at appropriate positions, e.g. $\tau \doteq [\sigma_1] w_1 w_2 [\sigma_3] w_4 w_5 [\sigma_6]$.

In general, a sequence of flaws will be called a *word*, and a sequence of flaws together with some intermediate states (such as $[\sigma_1] w_1 w_2 [\sigma_3] w_4 w_5 [\sigma_6]$) will be called a *pattern*. For a pattern $X$ we define $\langle X \rangle = \{\tau \mid \tau \doteq X\}$ to be the set of walks consistent with $X$. The length of $X$ (i.e. the number of flaws in it) is denoted as $|X|$.

**Lemma 10.** *For any word $W$ and state $\sigma$ we have*

$$\sum_{\tau \in \langle W[\sigma] \rangle} p(\tau) \quad \leq \quad \gamma^{\mathtt{init}} \cdot \lambda_W \cdot \omega(\sigma) \qquad (8a)$$

$$\sum_{\tau \in \langle W \rangle} p(\tau) \quad \leq \quad \gamma^{\mathtt{init}} \cdot \lambda_W \qquad (8b)$$

*where for a word $W = w_1 \ldots w_t$ we denoted $\lambda_W = \prod_{i \in [t]} \lambda_{w_i}$. (As described in the previous paragraph, $\langle W[\sigma] \rangle$ is the set of walks $\tau$ whose sequence of flaws is $W$ and the last state is $\sigma$.)*

*Proof.* Summing (8a) over $\sigma \in \Omega$ gives (8b), so it suffices to prove the former inequality. We use induction on the length of $W$. If $W$ is empty then $\langle W[\sigma] \rangle$ contains a single walk $\tau$ with the state $\sigma$; we then have $p(\tau) = \omega^{\mathtt{init}}(\sigma)$, and the claim follows from the definition of $\gamma^{\mathtt{init}}$ in Theorem 7. This establishes the base case. Now consider a word $W' = Wf$

with $f \in F$ and a state $\sigma'$. We can write

$$\begin{aligned}\sum_{\tau'\in\langle W'[\sigma']\rangle} p(\tau') &= \sum_{\sigma:\sigma'\in A(f,\sigma)} \sum_{\tau\in\langle W[\sigma]\rangle} p(\tau)\cdot\rho(\sigma'|f,\sigma) \\ &\overset{(a)}{\le} \sum_{\sigma:\sigma'\in A(f,\sigma)} \gamma^{\texttt{init}}\cdot\lambda_W\cdot\omega(\sigma)\cdot\rho(\sigma'|f,\sigma) \\ &\overset{(b)}{\le} \gamma^{\texttt{init}}\cdot\lambda_W\cdot[\lambda_f\cdot\omega(\sigma')] \\ &= \gamma^{\texttt{init}}\cdot\lambda_{W'}\cdot\omega(\sigma')\end{aligned}$$

where (a) is by the induction hypothesis, and (b) follows from (2a). This gives the induction step, and thus concludes the proof of the lemma. ☐

Next, we need define *stable sequences* and *stable walks*.

**Definition 11.** *A sequence of sets $\varphi = (I_1, \ldots, I_s)$ with $s \ge 1$ is called* stable *if $I_r \in \texttt{Ind}(F)$ for each $r \in [s]$ and $I_{r+1} \subseteq \Gamma^+(I_r)$ for each $r \in [s-1]$.*

**Definition 12.** *A word $W = w_1 \ldots w_t$ is called* stable *if it can be partitioned into non-empty words as $W = W_1 \ldots W_s$ such that flaws in each word $W_r$ are distinct, and the sequence $(I_1, \ldots, I_s)$ is stable where $I_r$ is the set of flaws in $W_r$ (for $r \in [s]$). If in addition each word $W_r = w_i \ldots w_j$ satisfies $w_i \prec_\pi \ldots \prec_\pi w_j$ then $W$ is called* $\pi$-stable.

*A walk $\tau \doteq W$ is called stable ($\pi$-stable) if the word $W$ is stable ($\pi$-stable).*

It can be shown that for a stable word the partitioning in Definition 12 is unique. Let $\texttt{Stab}_\pi$ be the set of $\pi$-stable words $W$ that satisfy the following condition:

- *There exists walk $\tau$ such that either $\tau \doteq W$ or $\tau \doteq \texttt{REV}[W]$, where $\texttt{REV}[W] = w_t \ldots w_1$ is the reverse of word $W = w_1 \ldots w_t$.*

For a stable word $W$ let $R_W$ be the first set (the "root") of the stable sequence $\varphi = (I_1, \ldots, I_s)$ corresponding to $W$, i.e. $R_W = I_1$. (If $W$ is empty then $R_W = \varnothing$). Denote $\texttt{Stab}_\pi(R) = \{W \in \texttt{Stab}_\pi : R_W = R\}$, $\texttt{Stab}_\pi(t) = \{W \in \texttt{Stab}_\pi : |W| \ge t\}$ and $\texttt{Stab}_\pi(R,t) = \texttt{Stab}_\pi(R) \cap \texttt{Stab}_\pi(t)$. The following result is proven in [12] using techniques from [18], [1].

**Theorem 13.** *Suppose that $(\rho, \sim)$ satisfies either the cluster expansion condition* (2b) *or the Shearer's condition from Definition 6. Then*

$$\sum_{W\in\texttt{Stab}_\pi(R,t)} \lambda_W \le \mu(R)\cdot\theta^t \qquad \forall R \in \texttt{Ind}(F) \qquad (9)$$

**Commutativity** From now on we assume that $(\rho, \sim)$ is weakly commutative. Therefore, for any walk $\tau = \ldots \sigma_1 \xrightarrow{f} \sigma_2 \xrightarrow{g} \sigma_3 \ldots$ with $f \not\cong g$ there exists another walk $\tau' = \ldots \sigma_1 \xrightarrow{g} \sigma_2' \xrightarrow{f} \sigma_3 \ldots$ obtained from $\tau$ by applying the SWAP operator to the subwalk $\sigma_1 \xrightarrow{f} \sigma_2 \xrightarrow{g} \sigma_3$. Such operation will be called a *valid swap* applied to $\tau$. A mapping $\Phi$ on a set of walks that works by applying some sequence of valid swaps will be called a *swapping mapping*. Note that if $\tau' = \Phi(\tau)$ then the first and the last states of $\tau'$ coincide with that of $\tau$, and $\lambda_{W'} = \lambda_W$ where $\tau \doteq W$, $\tau' \doteq W'$. Furthermore, if $(\rho, \sim)$ is strongly commutative then $p(\tau') = p(\tau)$.

We now deal with the case when $\Lambda$ is an arbitrary deterministic strategy, and so walks $\tau \in \texttt{Bad}(t)$ are not necessarily $\pi$-stable. Our approach will be to construct a *bijective* swapping mapping $\Phi$ that sends walks $\tau \in \texttt{Bad}(t)$ to some canonical walks, namely either to $\pi$-stable walks (which will work only in the atomic case) or to the *reverse* of such walks (which will work in the general case).

**Proof of Theorem 7(b)** Assume that $(\rho, \sim)$ is atomic. This gives the following observation.

**Proposition 14** ([7])**.** *Walk $\tau = \sigma_1 \xrightarrow{w_1} \sigma_2 \ldots \sigma_t \xrightarrow{w_t} \sigma_{t+1}$ can be uniquely reconstructed from the sequence of flaws $w_1 \ldots w_t$ and the final state $\sigma_{t+1}$.*

*Proof.* By atomicity, state $\sigma_i$ can be uniquely reconstructed from the flaw $w_i$ and the state $\sigma_{i+1}$. Applying this argument for $i = t, t-1, \ldots, 1$ gives the claim. ☐

The proposition allows us to write walks more compactly as $\tau = w_1 \ldots w_t[\sigma_{t+1}]$. Also, Lemma 10 gives for a walk $\tau = W[\sigma_{t+1}]$ that $p(\tau) \le \gamma^{\texttt{init}} \cdot \lambda_W \cdot \omega(\sigma_{t+1})$. In [12] we prove

**Theorem 15.** *Suppose that $(\rho, \sim)$ is atomic and weakly commutative. There exists a set of $\pi$-stable walks $\texttt{Bad}_\pi(t)$ and a swapping mapping $\Phi : \texttt{Bad}(t) \to \texttt{Bad}_\pi(t)$ which is a bijection.*

We can now prove Theorem 7(b):

$$\begin{aligned}Pr[\#\text{steps} \ge t] &= \sum_{\tau\in\texttt{Bad}(t)} p(\tau) \le \sum_{\tau=W[\sigma]\in\texttt{Bad}(t)} \gamma^{\texttt{init}}\cdot\lambda_W\cdot\omega(\sigma) \\ &\overset{(a)}{=} \sum_{\tau=W[\sigma]\in\texttt{Bad}_\pi(t)} \gamma^{\texttt{init}}\cdot\lambda_W\cdot\omega(\sigma) \\ &\overset{(b)}{\le} \gamma^{\texttt{init}}\cdot\sum_{R\in\texttt{Ind}^{\texttt{init}}} \mu(R)\cdot\theta^t = \theta^{t-T}\end{aligned}$$

where (a) follows from Theorem 15, (b) can be derived from Theorem 13, and constant $T$ is defined in Theorem 7(b).

**Reverse stable sequences** To prove Theorem 7(c) and Theorem 8, we will use *reverse stable sequences* instead of *forward stable sequences*.

Walk $\tau$ will be called a *prefix* of a walk $\tau'$ if $\tau'$ starts with $\tau$. Walk $\tau$ is a *proper prefix* of $\tau'$ if in addition $\tau' \neq \tau$. A word $W$ is called a prefix of $\tau$ if $\tau \doteq WU$ for some word $U$.

A set of walks $\mathcal{X}$ will be called *valid* if (i) all walks in $\mathcal{X}$ follow the same deterministic strategy (not necessarily the one used in Algorithm 1), and (ii) for any $\tau, \tau' \in \mathcal{X}$ the walk $\tau$ is not a proper prefix of $\tau'$.

For a walk $\tau$ containing flaw $f$ we define word $W_\tau^f$ as the longest prefix of $\tau$ that ends with $f$. Thus, we have $\tau \doteq W_\tau^f U$ where $W_\tau^f = \ldots f$ and word $U$ does not contain $f$. We will also allow $f = \varnothing$; in this case we say that any walk $\tau$ contains such $f$, and define word $W_\tau^f$ so that $\tau \doteq W_\tau^f$. Recall that for a word $W = w_1 \ldots w_t$ its reverse is denoted as $\texttt{REV}[W] =$

$w_t \dots w_1$. The following result proved in the full version of the paper [12] is probably the most technical part of the proof.

**Theorem 16.** *Fix $f \in F \cup \{\varnothing\}$, and let $\mathcal{X}^f$ be a valid set of walks containing $f$. If $(\rho, \sim)$ is weakly commutative then there exists a set of walks $\mathcal{X}^f_\pi$ and a swapping mapping $\Phi^f : \mathcal{X}^f \rightarrow \mathcal{X}^f_\pi$ which is a bijection such that*
*(a) for any $\tau \in \mathcal{X}^f_\pi$ the word $W = \mathtt{REV}[W^f_\tau]$ is $\pi$-stable ($W \in \mathtt{Stab}_\pi$), and $R_W = \{f\}$ if $f \in F$;*
*(b) for any word $W$ the set $\{\tau \in \mathcal{X}^f_\pi \mid \mathtt{REV}[W^f_\tau] = W\}$ is valid.*

**Proof of Theorem 7(c)** In this case we have $\mathtt{Ind}^{\mathtt{init}} = \mathtt{Ind}(F)$. We will use Theorem 16 with $f = \varnothing$ and $\mathcal{X}^f = \mathtt{Bad}(t)$. Part (a) gives that for any $\tau \stackrel{\bullet}{=} W'$ from $\mathcal{X}^\varnothing_\pi$ the word $W = \mathtt{REV}[W']$ satisfies $W \in \mathtt{Stab}_\pi(R, t)$ for some $R \in \mathtt{Ind}(F)$. We can write

$$Pr[\#\text{steps} \geq t] \;=\; \sum_{\tau \in \mathtt{Bad}(t)} p(\tau) \stackrel{\text{(a)}}{=} \sum_{\tau \in \mathcal{X}^\varnothing_\pi} p(\tau) \;\leq\; \theta^{t-T}$$

where in (a) we used bijectiveness of mapping $\Phi^\varnothing$ and strong commutativity of $(\rho, \sim)$, and the rest is similar to the derivation above.

**Analysis of the parallel algorithm: Proof of Theorem 8** This case requires dealing with "partial execution logs": we need to take sums over walks $\tau \stackrel{\bullet}{=} W$ for which word $W$ contains a certain subword. Our approach is to "move" this subword to the beginning of the walk via valid swaps, and then use the following result (which is proved by induction on the combined total length of walks in $\mathcal{X}$).

**Theorem 17.** *Consider a word $W$ and a valid set of walks $\mathcal{X}$ such that $W$ is a prefix of every walk in $\mathcal{X}$. Then*

$$\sum_{\tau \in \mathcal{X}} p(\tau) \leq \gamma^{\mathtt{init}} \cdot \lambda_W \tag{10}$$

*Proof.* We use induction on $\sum_{\tau \in \mathcal{X}} (|\tau| - |W|)$. The base case $\sum_{\tau \in \mathcal{X}} (|\tau| - |W|) = 0$ is straightforward: we then have $\mathcal{X} \subseteq \langle W \rangle$, and so the claim follows from Lemma 10. Consider a valid set $\mathcal{X}$ with $\sum_{\tau \in \mathcal{X}} (|\tau| - |W|) \geq 1$. Let $\hat{\tau}$ be a longest walk in $\mathcal{X}$, then $|\hat{\tau}| \geq |W| + 1$. Let $\hat{\tau}^-$ be the proper prefix of $\hat{\tau}$ of length $|\hat{\tau}| - 1$. We have $\hat{\tau}^- \notin \mathcal{X}$ since $\mathcal{X}$ is a valid set. Define set $\mathcal{Y}$ as follows: $\mathcal{Y} = \{\tau \in \mathcal{X} \mid \hat{\tau}^- \text{ is a proper prefix of } \tau\}$. By the choice of $\hat{\tau}$ we get $|\tau| = |\hat{\tau}|$ for all $\tau \in \mathcal{Y}$, and so we must have $\tau = \hat{\tau}^- \xrightarrow{w} \sigma$ for some $w \in F$ and $\sigma \in \Omega$. Since all walks in $\mathcal{X}$ follow the same deterministic strategy, the flaw $w$ in the expression $\tau = \hat{\tau}^- \xrightarrow{w} \sigma$ must be the same for all $\tau \in \mathcal{Y}$. Thus, $\mathcal{Y} = \{\hat{\tau}^- \xrightarrow{w} \sigma \mid \sigma \in Y\}$ for some set of flaws $Y \subseteq F$. In fact, we must have $Y \subseteq A(w, \hat{\sigma})$ where $\hat{\sigma}$ is the final state of $\hat{\tau}^-$.

Define $\mathcal{X}^- = (\mathcal{X} - \mathcal{Y}) \cup \{\hat{\tau}^-\}$. We have

$$\begin{aligned} \sum_{\tau \in \mathcal{X}} p(\tau) - \sum_{\tau \in \mathcal{X}^-} p(\tau) \;&=\; \sum_{\tau \in \mathcal{Y}} p(\tau) - p(\hat{\tau}^-) \\ &=\; p(\hat{\tau}^-) \cdot \left[ \sum_{\sigma \in Y} \rho(\sigma | w, \hat{\sigma}) - 1 \right] \leq 0 \end{aligned}$$

It is straightforward to check that set $\mathcal{X}^-$ is valid, and $W$ is a prefix of every walk in $\mathcal{X}^-$. Using the induction hypothesis for $\mathcal{X}^-$ and the inequality above gives the claim for $\mathcal{X}$. $\square$

To elaborate the argument, consider executions of Algorithm 2 consisting of at least $s$ rounds. For each such execution let $\tau$ be the walk containing flaws addressed in the first $s-1$ rounds and the first flaw addressed in round $s$. Let $\mathtt{BadPar}(s)$ be the set of such walks $\tau$. We prove that each $\tau \in \mathtt{BadPar}(s)$ contains a "chain" of length $s$, i.e. a subsequence $u_1 \dots u_s$ satisfying $u_i \cong u_{i+1}$ for $i \in [s-1]$. We can write $\mathtt{BadPar}(s) = \bigcup_{f \in F} \mathtt{BadPar}^f(s)$ where $\mathtt{BadPar}^f(s)$ is the set of those walks in $\mathtt{BadPar}(s)$ that contain a chain of length $s$ ending with $f$. We now apply Theorem 16 with the set $\mathcal{X}^f = \mathtt{BadPar}^f(s)$, and get $\sum_{\tau \in \mathtt{BadPar}^f(s)} p(\tau) = \sum_{\tau \in \mathcal{X}^f_\pi} p(\tau)$. Every walk in $\mathcal{X}^f_\pi$ starts with a prefix of the form $\mathtt{REV}[W]$ for some $W \in \mathtt{Stab}_\pi(\{f\}, s)$. By applying Theorem 17 for every such prefix we obtain an upper bound on $\sum_{\tau \in \mathcal{X}^f_\pi} p(\tau)$, and then complete the proof using the same arguments as before.


## Acknowledgment

The author is supported by the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no 616160.


## References


[1] N. Harvey and J. Vondrák, "An algorithmic proof of the Lovász local lemma via resampling oracles," in *FOCS*, 2015.
[2] D. Achlioptas and F. Iliopoulos, "Focused stochastic local search and the Lovász local lemma," in *SODA*, 2016.
[3] R. A. Moser and G. Tardos, "A constructive proof of the general Lovász local lemma," *J. ACM*, vol. 57, no. 2, 2010.
[4] P. Erdős and L. Lovász, "Problems and results on 3-chromatic hypergraphs and some related questions," in *Colloq. Math. Soc. J. Bolyai, Infinite and Finite Sets*. North-Holland, 1975, vol. 10, pp. 609–627.
[5] M. Szegedy, "The Lovász local lemma - a survey," in *Computer Science Theory and Applications*, A. A. Bulatov and A. M. Shur, Eds. Springer, Lecture Notes in Computer Science, 2013, vol. 7913, pp. 1–11.
[6] D. G. Harris and A. Srinivasan, "A constructive algorithm for the Lovász local lemma on permutations," in *SODA*, 2014, pp. 907–925.
[7] D. Achlioptas and F. Iliopoulos, "Random walks that find perfect objects and the Lovász local lemma," in *55th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, 2014, pp. 494–503.
[8] P. Erdős and J. Spencer, "The lopsided Lovász local lemma and latin transversals," *Discrete Applied Mathematics*, vol. 30, pp. 151–154, 1991.
[9] L. Lu, A. Mohr, and L. Székely, "Quest for negative dependency graphs," in *Recent Advances in Harmonic Analysis and Applications*, D. Bilyk, L. De Carli, A. Petukhov, A. Stokolos, and B. Wick, Eds. Springer, 2013.
[10] A. Mohr, "Applications of the lopsided Lovász local lemma regarding hypergraphs," Ph.D. dissertation, University of South Carolina, 2013.
[11] D. Knuth, "The art of computer programming, Volume 4B (draft, prefascicle 6a)," http://web.archive.org/web/20150815030301/http://www-cs-faculty.stanford.edu/ uno/news.html, Jul. 2015.
[12] V. Kolmogorov, "Commutativity in the algorithmic Lovász local lemma," *CoRR*, vol. abs/1506.08547v7, Aug. 2016.
[13] R. Bissacot, R. Fernández, A. Procacci, and B. Scoppola, "An improvement of the Lovász local lemma via cluster expansion," *Combin. Probab. Comput.*, vol. 20, pp. 709–719, 2011.
[14] W. Pegden, "An extension of the Moser-Tardos algorithmic local lemma," *SIAM Journal on Discrete Mathematics*, vol. 28(2), pp. 911–917, 2014.
[15] J. Böttcher, Y. Kohayakawa, and A. Procacci, "Properly coloured copies and rainbow copies of large graphs with small maximum degree," *Random Structures and Algorithms*, vol. 40(4), pp. 425–436, 2012.

[16] S. Ndreca, A. Procacci, and B. Scoppola, “Improved bounds on coloring of graphs,” *European Journal of Combinatorics*, vol. 33(4), pp. 592–609, 2012.

[17] J. B. Shearer, “On a problem of Spencer,” *Combinatorica*, vol. 5(3), pp. 241–245, 1985.

[18] K. B. R. Kolipaka and M. Szegedy, “Moser and Tardos meet Lovász,” in *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, ser. STOC, 2011, pp. 235–244.

[19] I. Giotis, L. Kirousis, K. I. Psaromiligkos, and D. M. Thilikos, “On the algorithmic Lovász local lemma and acyclic edge coloring,” in *Proceedings of ANALCO*, 2015.

[20] V. Kolmogorov, “Commutativity in the random walk formulation of the Lovász local lemma,” *CoRR*, vol. abs/1506.08547v1, Jun. 2015.