A New Perspective on Stochastic Local Search and the Lovász Local Lemma

Dimitris Achlioptas ∗
University of California Santa Cruz
optas@cs.ucsc.edu

Fotis Iliopoulos †
University of California Berkeley
fotis.iliopoulos@berkeley.edu

Alistair Sinclair ‡
University of California Berkeley
sinclair@berkeley.edu

Abstract

We present a new perspective on the analysis of stochastic local search algorithms via linear algebra. Our key insight is that LLL-inspired convergence arguments can be seen as a method for bounding the spectral radius of a matrix specifying the algorithm to be analyzed. Armed with this viewpoint we give a unified analysis of all entropy compression applications, connecting backtracking algorithms to the LLL in the same fashion that existing analyses connect resampling algorithms to the LLL. We then give a new convergence condition that seamlessly handles resampling algorithms that can detect, and back away from, unfavorable parts of the state space. We give several applications of this condition, notably a new vertex coloring algorithm for arbitrary graphs that uses a number of colors that matches the algorithmic barrier for random graphs. Finally, we introduce a generalization of Kolmogorov’s notion of commutative algorithms [52], cast as matrix commutativity, which affords much simpler proofs both of the original results and of recent extensions.

∗ Research supported by NSF grant CCF-1514434.
† Research supported by NSF grant CCF-1514434 and the Onassis Foundation.
‡ Research supported by NSF grant CCF-1514434.
1 Introduction

Numerous problems in computer science and combinatorics can be formulated as searching for objects lacking certain bad properties, or “flaws”. For example, constraint satisfaction problems like satisfiability and graph coloring can be seen as searching for objects (truth assignments, colorings) that are flawless, in the sense that they do not violate any constraint. A large class of algorithms for finding flawless objects employs “stochastic local search”; such algorithms start with a flawed object and try to make it flawless via small randomized changes that in each step focus on eradicating a specific flaw (while potentially introducing others). Given their great practical success, it is natural to ask whether there are conditions under which stochastic local search algorithms provably work efficiently, and use these conditions to show that interesting families of instances of hard problems are in fact tractable.

The Lovász Local Lemma (LLL) \cite{28} is a powerful tool for proving the existence of flawless objects that has had far-reaching consequences in computer science and combinatorics \cite{12, 55}. Roughly speaking, it asserts that, given a collection of a bad events in a probability space, if all of them are individually not too likely, and independent of most other bad events, then the probability that none of them occurs is strictly positive; hence a flawless object exists. For example, the LLL implies that every \( k \)-CNF formula in which each clause shares variables with fewer than \( 2^k/e \) other clauses is satisfiable. Remarkably, this is tight \cite{32}.

In groundbreaking work, Moser \cite{56}, joined by Tardos in \cite{57}, showed that a simple local search algorithm can be used to make the LLL constructive for product probability spaces. For example, the Moser-Tardos algorithm for satisfiability amounts to starting at a uniformly random truth assignment and, as long as violated clauses exist, selecting any such clause and resampling all of its variables uniformly at random. Following this work, a large amount of effort has been devoted to making all known variants of the LLL constructive: see, e.g., \cite{50, 51, 22, 60, 42, 4, 44, 5}.

Moser’s original analysis also inspired a parallel line of work centered on the entropy compression method. This method has been used primarily to analyze backtracking algorithms, e.g., for non-repetitive sequences \cite{36, 27}, acyclic edge coloring \cite{30}, non-repetitive list-coloring \cite{34}, the Thue choice number \cite{35}, and pattern avoidance \cite{59}. More recently, entropy compression was used to analyze resampling algorithms for stochastic control \cite{6} and graph list-coloring \cite{53}, in the latter case dramatically simplifying the celebrated result of Johansson \cite{47}. While the spirit of the analysis in all these works is close to \cite{56}, they are not derived from a known form of the LLL and indeed often improve on earlier results obtained from the LLL. Instead, the fact that the algorithm under consideration reaches a flawless object is established in each case by a problem-specific counting argument.

In this paper we introduce a new viewpoint for the analysis of local search algorithms, based on linear algebra. Our key insight is the following:

\textit{LLL-inspired convergence arguments can be seen as a method for bounding the spectral radius of a matrix specifying the algorithm to be analyzed.}

Among the benefits of this new viewpoint, which we will present in a moment, are the following:

- A unified analysis of all entropy compression applications, connecting backtracking algorithms to the LLL in the same fashion that existing analyses connect resampling algorithms to the LLL.
- A new convergence condition that seamlessly handles resampling algorithms that can detect, and back away from, unfavorable parts of the state space.
- Several applications of this condition, notably a new vertex coloring algorithm for arbitrary graphs that uses a number of colors that matches the algorithmic barrier for random graphs. Thus, any improvement on our algorithm’s guarantee requires a breakthrough in random graph theory.
- A generalization of Kolmogorov’s notion of commutative algorithms \cite{52}, cast as matrix commutativity, which affords much simpler proofs both of the original results and of recent extensions.
1.1 The Lovász Local Lemma as a Spectral Condition

Let $\Omega$ be a (large) finite set of objects and let $\Omega^* \subseteq \Omega$ be the “bad” part of $\Omega$, comprising the flawed objects; e.g., for a CNF formula on $n$ variables $\Omega = \{0, 1\}^n$ and $\Omega^*$ comprises all non-satisfying assignments. Imagine a particle trying to escape $\Omega^*$ by following a Markov chain$^1$ on $\Omega$ with transition matrix $P$. Our task is to develop conditions under which the particle eventually escapes, thus establishing in particular that $\Omega^* \neq \Omega$. (Motivated by this view, we also refer to objects as states.) Letting $A$ be the $|\Omega^*| \times |\Omega^*|$ submatrix of $P$ that corresponds to transitions from $\Omega^*$ to $\Omega^*$, and $B$ the submatrix that corresponds to transitions from $\Omega^*$ to $\Omega \setminus \Omega^*$, we see that, after a suitable permutation of its rows and columns, $P$ can be written as

$$P = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}.$$ 

Here $I$ is the identity matrix, since we assume that the particle stops after reaching a flawless state.

Let $\theta = [\theta_1 | \theta_2]$ be the row vector that corresponds to the probability distribution of the starting state, where $\theta_1$ and $\theta_2$ are the vectors that correspond to states in $\Omega^*$ and $\Omega \setminus \Omega^*$, respectively. Then, the probability that after $t$ steps the particle is still inside $\Omega^*$ is exactly $\|\theta_1 A^t\|_1$. Therefore, for any initial distribution $\theta$, the particle escapes $\Omega^*$ if and only if the spectral radius, $\rho(A)$, of $A$ is strictly less than 1. Moreover, the rate of convergence is dictated by $1 - \rho(A)$. Unfortunately, since $A$ is huge and defined implicitly by an algorithm, the magnitude of its largest eigenvalue, $\rho(A)$, is not readily available.

In linear systems analysis, to sidestep the inaccessibility of the spectral radius, $\rho(A)$, one typically bounds instead some operator norm $\| \cdot \|$ of the matrix $A$, since $\rho(A) \leq \| A \|$ for any such norm. (For brief background on matrix norms see Appendix A.) Moreover, instead of bounding an operator norm of $A$ itself, one often first performs a “change of basis” $A' = MAM^{-1}$ and bounds $\| A' \|$, justified by the fact that $\rho(A) = \rho(A') \leq \| A' \|$, for any invertible matrix $M$. The purpose of the change of basis is to cast $A$ “in a good light” in the eyes of the chosen operator norm, in the hope of minimizing the cost of replacing the spectral norm with an operator norm. To demonstrate this approach in action, we start by showing how it captures the classical potential function argument.

Consider any function $\phi$ on $\Omega$ such that $\phi(\sigma) > 0$ for $\sigma \in \Omega^*$, while $\phi(\sigma) = 0$ for $\sigma \notin \Omega^*$. In our $k$-SAT example, $\phi(\sigma)$ could be the number of violated clauses under $\sigma$. The potential argument asserts that eventually $\phi = 0$ (i.e., the particle escapes $\Omega^*$) if $\phi$ is always reduced in expectation, i.e., if for every $\sigma \in \Omega^*$,

$$\sum_{\sigma' \in \Omega} P[\sigma, \sigma'] \phi(\sigma') < \phi(\sigma).$$

To express this argument via matrix norms, let $A' = MAM^{-1}$ where $M$ is the diagonal $|\Omega^*| \times |\Omega^*|$ matrix $\text{diag}(1/\phi(\sigma))$. Thus, $A'[\sigma, \sigma'] = A[\sigma, \sigma']\phi(\sigma')/\phi(\sigma)$. Recalling that $\| \cdot \|_\infty$ is the maximum row sum of a matrix, we see that the potential argument’s condition is nothing other than $\| A' \|_\infty < 1$.

Our starting point is the observation that all entropy compression arguments, and indeed all arguments in the algorithmic LLL literature, can be seen as dual to the potential function argument. That is, after a suitable change of basis $A' = MAM^{-1}$, they bound not $\| A' \|_\infty$, as the potential argument, but the dual norm $\| A' \|_1$. As a concrete demonstration, let us consider the Moser-Tardos algorithm for a $k$-CNF formula on $n$ variables with clauses $c_1, \ldots, c_m$, under the uniform measure on $\Omega = \{0, 1\}^n$. For simplicity, assume that the lowest indexed violated clause is always resampled, so that the state evolves as a Markov chain.

For each clause $c_i$, let $A_i$ be the $|\Omega^*| \times |\Omega^*|$ submatrix of $A$ comprising all rows (states) where the resampled clause is $c_i$. (All other rows of $A_i$ are 0). For $t \geq 1$, let $W_t$ contain every $t$-sequence of (indices of) clauses that has non-zero probability of comprising the first $t$ clauses resampled by the algorithm. In

$^1$Our framework does not require the state evolution to be Markovian, but we make this assumption here to simplify exposition.
other words, \( W_t \) is the set of all \( t \)-sequences of indices from \([m]\) corresponding to non-vanishing \( t \)-products of matrices from \( \{A_1, \ldots, A_m\} \), i.e., \( W_t = \{W = (w_i) \in [m]^t : \prod_{i=1}^t A_{w_i} \neq 0\} \). With these definitions, the first inequality below follows from the fact that \( \rho(A) \leq \|A\| \) for any operator norm \( \|\cdot\| \), the triangle inequality gives the second inequality and, crucially, the submultiplicativity of operator norms gives the third:

\[
\rho(A)^t = \rho(A^t) \leq \|A^t\| = \left\| \left( \sum_{i \in [m]} A_i \right)^t \right\| = \left\| \sum_{W \in W_t} \prod_{i=1}^t A_{w_i} \right\| \leq \sum_{W \in W_t} \left\| \prod_{i=1}^t A_{w_i} \right\| \leq \sum_{W \in W_t} \|A_{w_i}\|,
\]

(2)

Observe that (2) holds for every operator norm. To get a favorable bound here, we will apply (2) with the norm \( \|\cdot\|_1 \), i.e., the maximum column sum. We see that for all \( j \in [m] \), every column of \( A_j \) has at most one non-zero entry, since \( A_j(\sigma, \sigma') > 0 \) only if \( \sigma \) is the mutation of \( \sigma' \) so that \( c_j \) is violated. Recalling that all non-zero entries of \( A \) equal \( 2^{-k} \), we conclude \( \|A_j\|_1 = 2^{-k} \) for all \( j \in [m] \). Therefore, \( \|A^t\|_1 \leq |W_t|2^{-kt} \).

To bound \( |W_t| \) we use a simple necessary condition for membership in \( W_t \) which, by a standard counting argument, implies that if each clause shares variables with at most \( \Delta \) other clauses, then \( |W_t| \leq 2^m(e\Delta^2)^t \). Therefore, \( \rho(A)^t \leq 2^m(e\Delta^2)^t \) implying that if \( \Delta < 2^k/e \), then \( 1 > \|A\|_1 \geq \rho(A) \) and the algorithm terminates within \( O(m) \) steps with high probability.

1.2 Informal Statement of our Main Theorem

The matrix-norm perspective we introduce in this paper allows us not only to cast the probabilistic method aspect of the algorithmic LLL as a change of basis, and the overall approach as a dual potential function argument, but, more importantly, to significantly expand and refine the analysis, so that it can avoid a hard notion of dependence, i.e., a dependency or causality graph. This is because, unlike past works, our condition quantifies point-to-set correlations, so that interactions can be arbitrarily dense as long as they are sufficiently weak. Before stating our result we need to fix some notation.

Let \( \Omega \) be a discrete state space, and let \( F = \{f_1, f_2, \ldots, f_m\} \) be a collection of subsets (flaws) of \( \Omega \) such that \( \bigcup_{i \in [m]} f_i = \Omega^* \). For a state \( \sigma \), we denote by \( U(\sigma) = \{j \in [m] : f_j \ni \sigma\} \) the set of (indices of) flaws present in \( \sigma \). (Here and elsewhere, we shall blur the distinction between flaws and their indices.) We consider algorithms which, in each flawed state \( \sigma \in \Omega^* \), choose a flaw \( f_i \) in \( U(\sigma) \) and attempt to leave (“fix”) \( f_i \), by moving to a new state \( \sigma' \) according to a probability distribution \( \rho_i(\sigma, \cdot) \). We make minimal assumptions about how the algorithm chooses which flaw to address in each step, e.g., it will be enough for the algorithm to choose the flaw with lowest index according to some fixed permutation. (We discuss this point further in the formal statement of our results.) We refer to an attempt to fix a flaw, successful or not, as addressing it. We say that a transition \( \sigma \rightarrow \sigma' \), made to address flaw \( f_i \), introduces flaw \( f_j \in U(\sigma') \) if \( f_j \notin U(\sigma) \) or if \( j = i \). (Thus, a flaw (re)introduces itself when a transition fails to address it.)

Let us define for every flaw \( f_i \), \( i \in [m] \), and every set of (indices of) flaws \( S \subseteq [m] \), the matrix \( A_S^\sigma \) to be the \( |\Omega| \times |\Omega| \) matrix having a non-zero entry \( A_S^\sigma[\sigma, \sigma'] = \rho_i(\sigma, \sigma') \) for every transition \( \sigma \rightarrow \sigma' \) such that the set of flaws introduced contains \( S \).

**Theorem 1.1.** Let \( M \) be any \( |\Omega^*| \times |\Omega^*| \) invertible matrix. Let \( \|\cdot\| \) be any operator norm. For \( i \in [m] \) and \( S \subseteq [m] \), let \( \gamma_i^S := \|MA_S^\sigma M^{-1}\| \). If there exist positive real numbers \( \{\psi_i\}_{i=1}^m \) such that for all \( i \in [m] \),

\[
\frac{1}{\psi_i} \sum_{S \subseteq [m]} \gamma_i^S \prod_{j \in S} \psi_j < 1 ,
\]

then a local search algorithm as above reaches a flawless object quickly with high probability.
We quantify the phrase “quickly with high probability” in our formal statement of results. In applications, this typically means that the probability that the algorithm takes more than $sT$ steps is $\exp(-\Theta(s))$, for some $T$ that is linear in the size of the input.

We refer to the norm $\gamma_i^S := \|MA_i^S M^{-1}\|$ appearing in the above theorem as the charge of the pair $(i, S)$. These charges will play a crucial role in our analysis.

In Section 2.4 we will introduce a significant strengthening of Theorem 1.1 that, under certain conditions, allows us to sparsify each matrix $A_i^S$, thus reducing its norm, by zeroing out some or all entries that correspond to transitions where a superset of $S$ was introduced. Before doing so, we first discuss how Theorem 1.1 already captures and generalizes previous work on the algorithmic LLL.

## 2 Comparison with Previous LLL Conditions

Here we give background on the LLL and explain how our main theorem compares with previous algorithmic LLL conditions. In the interest of space, some additional background material appears in Appendix B.

### 2.1 Non-constructive Conditions

We start by stating the strongest form of the LLL that holds for arbitrary probability spaces and families of bad events (see e.g., [53, p.228]).

**General LLL.** Let $(\Omega, \mu)$ be a probability space and let $B = \{B_1, B_2, \ldots, B_m\}$ be a set of $m$ (bad) events. For each $i \in [m]$, let $L(i) \subseteq [m] \setminus \{i\}$ be such that $\mu(B_i \setminus \bigcap_{S \subseteq [m] \setminus (L(i) \cup \{i\})} B_j) \leq b_i$ for every $S \subseteq [m] \setminus (L(i) \cup \{i\})$. If there exist positive real numbers $\{\psi_i\}_{i=1}^m$ such that for all $i \in [m]$, 

$$\frac{b_i}{\psi_i} \sum_{S \subseteq \bigcup_{i \in L(i)} \bigcup_{j \in S} \psi_j \leq 1,$$

then the probability that none of the events in $B$ occurs is at least $\prod_{i=1}^m 1/(1 + \psi_i) > 0$.

Writing $x_i = \psi_i/(1 + \psi_i) \in [0, 1]$, condition (4) takes the more familiar form $b_i \leq x_i \prod_{j \in L(i)} (1 - x_j)$. The form in (4), though, is more amenable to refinement and comparison. The directed graph on $[m]$ where each vertex $i$ points to the vertices in $L(i)$ is known as the lopsided dependency graph.

The above form of the LLL is motivated by the fact that, in complex applications, small but non-vanishing correlations tend to travel arbitrarily far in the space $\Omega$. To isolate these dependencies so that they can be treated locally, it can be crucial [9, 26, 48, 49] to allow mild negative correlations between each bad event $B_i$ and the events outside its “special” set $L(i)$, achieved by allowing $b_i \geq \mu(B_i)$. The Lopsided LLL of Erdős and Spencer [29] corresponds to $b_i = \mu(B_i)$, i.e., to not allowing such negative correlations.

### 2.2 Constructive Conditions

Using the framework of Section 1.2 for local search algorithms, we say that flaw $f_i$ causes flaw $f_j$ and write $i \rightarrow j$, if there exist $\sigma \in f_i, \sigma' \in f_j$ such that $\rho_i(\sigma, \sigma') > 0$ and the transition $\sigma \rightarrow \sigma'$ introduces flaw $f_j$. (Thus, causality is to be interpreted as potential causality.) Let $\Gamma(i) = \{j : i \rightarrow j\}$ be the set of flaws caused by $f_i$. We call the digraph over $[m]$ in which each vertex $i$ points to the vertices in $\Gamma(i)$ the causality digraph.

Let $\mu$ be an arbitrary probability measure on $\Omega$. The distortion of $\mu$ associated with flaw $f_i$ is the greatest inflation of a state probability induced by sampling $\sigma \in f_i$ according to $\mu$ and addressing flaw $f_i$ at $\sigma$. More formally, for $i \in [m]$ and $\sigma' \in \Omega$, let $\nu_i(\sigma')$ be the probability of ending up in state $\sigma'$ after sampling a state
\[\sigma \in f_i\] according to \(\mu\), and then addressing \(f_i\) at \(\sigma\). The distortion associated with \(f_i\) is then

\[d_i := \max_{\sigma' \in \Omega} \frac{\nu_i(\sigma')}{\mu(\sigma')} \geq 1 .\]

If \(d_i = 1\), i.e., \(\nu_i(\sigma') = \mu(\sigma')\) for all \(\sigma' \in \Omega\), we say that the algorithm is a resampling oracle \([44]\) for \(f_i\).

Observe that a resampling oracle perfectly removes the conditioning on the old state \(\sigma\) belonging to \(f_i\), since the new state \(\sigma'\) is distributed according to \(\mu\). For example, if \(\mu\) is a product measure, this is precisely what is achieved by the resampling algorithm of Moser and Tardos \([57]\).

Requiring an algorithm to be a resampling oracle for every flaw may be impossible to achieve by local exploration within \(\Omega\), i.e., by "local search." (Note that restricting to local search is crucial since long-range resampling, even if it were possible, would tend to rapidly densify the causality digraph.) Allowing distortion frees the algorithm from the strict resampling requirement of perfect deconditioning. Optimizing the tradeoff between distortion and the density of the causality digraph has recently led to strong algorithmic results \([5, 53, 46, 31]\). (As we will see later, our results make this optimization task easier.)

**Algorithmic LLL.** Let \(\gamma_i := \mu(f_i)d_i\). If there exist positive real numbers \(\{\psi_i\}_{i=1}^m\) such that for all \(i \in [m]\),

\[\frac{\gamma_i}{\psi_i} \sum_{S \subseteq \Gamma(i)} \prod_{j \in S} \psi_j < 1 ,\]

then a local search algorithm as above reaches a flawless object quickly with high probability.

As shown in \([31]\), condition \(5\) is the algorithmic counterpart of the existential condition \(4\): a causality digraph is a lopsidependency graph for measure \(\mu\) with \(b_i = \gamma_i\) for all \(i \in [m]\). We include the proof of this fact from \([31]\) in Appendix C for completeness. In particular, when one has resampling oracles for all flaws, i.e., \(d_i = 1\) for all \(i \in [m]\), then condition \(4\) is the algorithmic counterpart of the Lopsided LLL, as established by Harvey and Vondrák \([44]\). Condition \(5\) also subsumes the flaws/actions condition of \([3]\): in that setting \(\rho_i(\sigma, \cdot)\) is uniform over the set of possible next states, while the analysis does not reference a measure \(\mu\). Taking \(\mu\) to be uniform and applying condition \(5\) in fact sharpens the condition of \([3]\).

We note that condition \(5\) can be improved in certain settings, i.e., under additional assumptions. Let \(G\) be an undirected graph on \([m]\) such that \(\Gamma(i)\) is a subset of the neighbors of \(i\), for every \(i \in [m]\). (One can trivially get such a \(G\) by ignoring the direction of arcs in the lopsidependency graph, but at the cost of potentially expanding the "neighborhood" of each vertex.) It was proven in \([5]\) that condition \(5\) can be replaced by the cluster expansion condition \([20]\) on \(G\), while in \([52]\) it was proven that condition \(5\) can be replaced by Shearer’s condition \([65]\). Both of these conditions benefit by restricting consideration to independent sets of \(G\) (see Appendix B). Also, Harris and Srinivasan \([39, 41]\) have developed improved conditions for the convergence of algorithms operating in the so-called variable setting \([57]\), based on refinements of the notion of dependency between bad events. These improvements are incomparable to condition \(5\), as they do not apply to general local search algorithms (for instance, all algorithms in the variable setting are commutative).

### 2.3 Our New Condition

Our Theorem \([1, 1]\) is a strict generalization of the above algorithmic LLL condition \(5\). To see this, observe that if \(M\) is the diagonal matrix \(\text{diag}(\mu(\sigma))\) and \(\| \cdot \| = \| \cdot \|_1\), then \(\gamma_i^S \leq \gamma_i^\emptyset = \gamma_i\) for every set \(S \subseteq [m]\), as

\[\gamma_i^S = \| MA_i^S M^{-1} \|_1 \leq \| MA_i^\emptyset M^{-1} \|_1 = \max_{\sigma' \in \Omega} \sum_{\sigma \in f_i} \frac{\mu(\sigma)}{\mu(\sigma')} \rho_i(\sigma, \sigma') = \max_{\sigma' \in \Omega} \frac{\nu(\sigma')}{\mu(\sigma')} \mu(f_i) = d_i \mu(f_i) = \gamma_i .\]

Hence, since also \(\gamma_i^S = 0\) for \(S \not\subseteq \Gamma(i)\), the l.h.s. of our condition \(5\) is never larger than the l.h.s. of \(5\).
As a quick example of where this may be helpful, let \( c_i \) be a clause in a CNF formula and let \( S \) be a set of clauses that share variables with \( c_i \) but which can never be violated simultaneously, e.g., because two of them disagree on the sign of a variable. Trivially, \( \gamma_i^S = 0 \), even though \( \gamma_i = \gamma_i^0 > 0 \) (assuming the algorithm ever addresses \( c_i \)). Clearly, the advantageous vanishing of \( \gamma_i^S \) here was due to a structural property of \( F \). In the absence of such a structural property, we may still be able to achieve \( \gamma_i^S = 0 \) by designing the algorithm so that it never transitions from a state where \( c_i \) is violated to a state where all clauses in \( S \) are violated.

Next we discuss a strengthening of Theorem 1.1 that leads to significant algorithmic improvements.

### 2.4 Refinement of Our Condition

Even though, as we have just seen, our Theorem 1.1 already improves upon all existing general algorithmic LLL conditions, we might hope to do even better. Observe that in Theorem 1.1 a matrix \( A_i^S \) has a non-zero entry whenever addressing flaw \( f_i \) introduces any superset \( S' \supseteq S \) of flaws. Ideally, we would like a non-zero entry in \( A_i^S \) only when the set of flaws introduced is exactly \( S \), so that the matrices \( A_i^S \) partition \( A_i \).

The reason for this apparent weakness is that flaws introduced by fixing \( f_i \) may later be fixed “collaterally,” i.e., as a result of addressing other flaws, rather than by being specifically addressed by the algorithm, so we cannot charge those flaws unambiguously to \( A_i^S \). While it may initially seem that the possibility of collateral fixes of flaws cannot be detrimental, from an analysis perspective they actually represent a loss of control over the progress of the algorithm. Consider for example a step in which addressing flaw \( f_i \) introduces a set of flaws, all of which end up being fixed collaterally. The analysis will charge \( \gamma_i^0 \) for this step, even though (had we been able to detect that that is what happened) we could have charged \( \gamma_i^S \).

Tracking collateral fixes and taking them into account not only wreaks havoc on theoretical bounds, but also appears to be a bad idea in practice [63, 64, 14, 15]: for example, focused local search satisfiability algorithms which select variables to flip based only on the flaws they introduce are known to fare much better than algorithms that weigh this damage against the benefit of the collaterally fixed clauses. As we will see, if an algorithm never makes collateral fixes, then we can sharpen Theorem 1.1 so that each matrix \( A_i^S \) has a non-zero entry only when the set of flaws introduced is exactly \( S \), as desired. This leads to a significant sparsification of the matrices, and a corresponding reduction of the charges \( \gamma_i^S \).

A natural class of local search algorithms with no collateral fixes are backtracking algorithms for CSPs. In these algorithms the state space \( \Omega \) is the set of all partial assignments that do not violate any constraint, while there is one flaw for each unassigned variable: if fixing a flaw (i.e., assigning a variable) causes one or more constraints to become violated, the algorithm backtracks by unassigning not only the last variable set but several more—typically all variables involved in some violated constraint. Examples of such algorithms include [36, 30, 27, 34, 35, 59, 61, 21, 53, 13, 19]. Our sharpened theorem immediately provides a unified and greatly simplified analysis of such algorithms (see Section 6 for examples). Note in particular that our ability to control point-to-set correlations in Theorem 1.1 is crucial for this: in principle, backtracking steps, by their nature, may introduce many flaws, but because this happens only in very specific circumstances, the associated charges \( \gamma_i^S \) are small.

Having developed an algorithmic local lemma for backtracking algorithms, we extend our framework to cover algorithms that make both “resampling” and backtracking steps. The key for this is to introduce the notion of a primary flaw, which is a flaw that, once present, can only be eradicated by being addressed by the algorithm (i.e., it cannot be fixed collaterally). Note that all flaws in a backtracking algorithm are primary. The strongest form of our theorem (Theorem 3.3 in the next section), which applies to arbitrary local search algorithms, allows us to restrict the non-zero entries of each matrix \( A_i^S \) to transitions where addressing flaw \( f_i \) introduces precisely the set of primary flaws in \( S \), as well as any superset of the non-primary flaws in \( S \). This form of the theorem is particularly powerful when analyzing resampling algorithms that include additional backtracking steps in order to retreat from “bad” parts of the state space. The reason is that, with the separation of flaws into primary and non-primary, the charge \( \gamma_i^0 \) reflects exactly the distortion.
due to transitions that “make progress”, i.e., that do not introduce primary flaws, and all of whose introduced non-primary flaws are fixed collaterally. Thus, if in some region of state space, the algorithm blows up \( \gamma_1^0 \), this is a signal that we should modify it to backtrack instead of pressing on. Even though doing so creates dependencies between potentially large sets of flaws, our capacity to quantify point-to-set correlations allows us to charge such steps in proportion to their frequency. We illustrate the power of this approach by adding backtracking steps to Molloy’s recent breakthrough resampling algorithm for coloring triangle-free graphs [53], in order to handle graphs with triangles.

3 Statement of Results

3.1 A New Algorithmic LLL Condition

Below we state our main result, which includes the strengthening of Theorem 1.1 discussed in Section 2.4.

**Definition 3.1** (Primary Flaws). A flaw \( f_i \) is primary if for every \( \sigma \in f_i \) and every \( j \neq i \), addressing \( f_j \) at \( \sigma \) always results in some \( \sigma' \in f_i \), i.e., \( f_i \) is never eradicated collaterally. For a given set \( S \subseteq [m] \), we write \( S^P \) and \( S^N \) to denote the indices that correspond to primary and non-primary flaws in \( S \), respectively.

**Definition 3.2** (Sparsified Matrices). For every \( i \in [m] \) and every set of flaw indices \( S \subseteq [m] \), let \( A_i^S \) be the \( |\Omega| \times |\Omega| \) matrix where \( A_i^S(\sigma, \sigma') = \rho_i(\sigma, \sigma') \) if the set of primary flaws introduced by the transition \( \sigma \rightarrow \sigma' \) equals \( S^P \) and the set of non-primary flaws introduced by \( \sigma \rightarrow \sigma' \) contains \( S^N \); otherwise \( A_i^S(\sigma, \sigma') = 0 \).

**Remark 3.1.** In Theorem [1.1] we used matrices where \( A_i^S(\sigma, \sigma') = \rho_i(\sigma, \sigma') \) if the set of flaws introduced by the transition \( \sigma \rightarrow \sigma' \) contained \( S = S^P \cup S^N \). The sparsification amounts to zeroing out all entries for which the set of primary flaws introduced is a strict superset of \( S^P \). In particular, if \( S^P = \emptyset \), then all entries corresponding to transitions that introduce primary flaws are zeroed-out.

For a state \( \sigma \), let \( e_\sigma \) denote the indicator vector of \( \sigma \), i.e., \( e_\sigma[\sigma] = 1 \) and \( e_\sigma[\tau] = 0 \) for all \( \tau \in \Omega \setminus \{\sigma\} \). The span of a probability distribution \( \theta : \Omega \rightarrow [0,1] \), denoted by \( \text{Span}(\theta) \), is the set of flaw indices that may be present in a state selected according to \( \theta \), i.e., \( \text{Span}(\theta) = \bigcup_{\sigma \in \Omega: \theta(\sigma) > 0} U(\sigma) \).

Let \( \pi \) be an arbitrary permutation over \( m \). We say that an algorithm follows the \( \pi \)-strategy if at each step it picks to address the flaw corresponding to the lowest index element of \( U(\sigma) \) according to \( \pi \).

**Theorem 3.3** (Main Result). Let \( M \) be any \( |\Omega^*| \times |\Omega^*| \) invertible matrix such that \( \sum_{\sigma \in \Omega} \| Me_\sigma \| = 1 \). Let \( \| \cdot \| \) be any operator norm. For every \( i \in [m] \) and \( S \subseteq [m] \), let \( \gamma_i^S = \| M A_i^S M^{-1} \| \). If there exist positive real numbers \( \{ \psi_i \}_{i \in [m]} \) such that for every \( i \in [m] \),

\[
\zeta_i := \frac{1}{\psi_i} \sum_{S \subseteq [m]} \gamma_i^S \prod_{j \in S} \psi_j < 1 ,
\]

then, for every permutation \( \pi \) over \([m]\), the probability that an algorithm following the \( \pi \)-strategy fails to find a flawless object within \((T_0 + s)/\delta \) steps is \( 2^{-s} \), where \( \delta = 1 - \max_{i \in [m]} \zeta_i \), and

\[
T_0 = \log_2 \| \theta^T M^{-1} \|_+ + \log_2 \left( \sum_{S \subseteq \text{Span}(\theta)} \prod_{j \in S} \psi_j \right) + \log_2 \left( \max_{S \subseteq [m]} \frac{1}{\prod_{j \in S} \psi_j} \right) ,
\]

where \( \| \cdot \|_+ \) denotes the dual norm of \( \| \cdot \| \).

To get a feeling for Theorem 3.3 we start by noting that in typical applications the sum in (6) is easily computable, as \( \gamma_i^S = 0 \) for the vast majority of subsets \( S \). Also, \( M \) will usually be a positive diagonal matrix \( \text{diag}(\mu(\sigma)) \), so that \( \sum_{\sigma \in \Omega} \| Me_\sigma \| = 1 \) means that \( \mu \) is a probability distribution. Thus, the vector \( \theta^T M^{-1} \) is the ratio of two probability distributions on \( \Omega \) so, typically, \( \log_2 \| \theta^T M^{-1} \|_+ = O(\log |\Omega|) \). In the important special case where, additionally, \( \| \cdot \| = \| \cdot \|_1 \), the time bound simplifies to the following.
Corollary 3.4. Let \( \mu > 0 \) be an arbitrary measure on \( \Omega \), let \( M \) be the \( |\Omega|^* \times |\Omega|^* \) matrix \( \text{diag}(\mu(\sigma)) \), and let \( \gamma^S_i = \|MA^S_i M^{-1}\|_1 \). If there exist positive real numbers \( \{\psi_i\}_{i \in [m]} \) such that for every \( i \in [m] \), condition (6) holds, then the conclusion of Theorem 3.3 holds with \( T_0 = \log_2 \mu_{\min}^{-1} + m \log_2 \left( \frac{1}{\gamma_{\min}} \right) \).

Remark 3.2. In applications of Corollary 3.4, typically, \( \{\psi_i\}_{i \in [m]} > 0 \) are such that \( T_0 = O(\log |\Omega| + m) \).

Remark 3.3. The requirement \( \sum_{\sigma \in \Omega} \|Me_\sigma\| = 1 \) is not necessarily. We impose it because in applications \( M \) is typically diagonal with positive entries, in which case the normalization \( \sum_{\sigma \in \Omega} \|Me_\sigma\| = 1 \) simplifies the expressions for the running time.

Remark 3.4. For any fixed permutation \( \pi \), the charges \( \gamma^S_i \) can be reduced by replacing \( A^S_i \) with the matrix \( A^S_i(\pi) \) that results by zeroing out every row \( \sigma \) of \( A^S_i \) for which \( i \) is not the lowest indexed element of \( U(\sigma) \) according to \( \pi \).

Remark 3.5. Theorem 3.5 holds also for algorithms using flaw choice strategies other than \( \pi \)-strategies. We discuss some such strategies in Section 4.3. However, there is good reason to expect that it does not hold for arbitrary flaw choice strategies, i.e., without additional assumptions (for more details see [52]).

3.2 Application to Graph Coloring

In the graph coloring problem one is given a graph \( G(V, E) \) and the goal is to find a mapping of \( V \) to a set of \( q \) colors so that no edge in \( E \) is monochromatic. The chromatic number, \( \chi(G) \), of \( G \) is the smallest integer \( q \) for which this is possible. Given a set \( \mathcal{L}_v \) of colors (called a list) for each vertex \( v \), a list-coloring maps each \( v \in V \) to a color in \( \mathcal{L}_v \) so that no edge in \( E \) is monochromatic. A graph is \( q \)-list-colorable if it has a list-coloring no matter how one assigns a list of \( q \) colors to each vertex. The list chromatic number, \( \chi_l(G) \), is the smallest \( q \) for which \( G \) is \( q \)-list-colorable. Clearly \( \chi_l(G) \geq \chi(G) \). A celebrated result of Johansson [47] established that there exists a large constant \( C > 0 \) such that every triangle-free graph with maximum degree \( \Delta \geq \Delta_0 \) can be list-colored using \( C\Delta/\ln \Delta \) colors. Very recently, using the entropy compression method, Molloy [53] improved Johansson’s result, replacing \( C \) with \( (1 + \epsilon) \) for any \( \epsilon > 0 \) and all \( \Delta \geq \Delta_\epsilon \). (Soon thereafter, Bernshtein [17] established the same bound for the list chromatic number, non-constructively, via [4], and Iliopoulos [46] showed that the algorithm of Molloy can be analyzed using [5].)

Our main result in this section is a generalization of Molloy’s result to graphs with a bounded number of triangles per vertex. Specifically, in Section 5 we establish the following general theorem for the list chromatic number (the triangle-free case corresponding to \( f = \Delta^2 + 1 \)).

Theorem 3.5. Let \( G \) be any graph with maximum degree \( \Delta \) in which the neighbors of every vertex span at most \( \Delta^2 / f \) edges. For all \( \epsilon > 0 \), there exists \( \Delta_\epsilon \) such that if \( \Delta \geq \Delta_\epsilon \), and \( f \in [\Delta^{\Delta_\epsilon \ln \Delta} \Delta^2, \Delta^2 + 1] \), then

\[
\chi_l(G) \leq (1 + \epsilon) \Delta / \ln \sqrt{f}.
\]

Furthermore, such a coloring can be found in polynomial time with high probability.

Theorem 3.5 is interesting for two reasons. First, random graphs suggest that it is sharp, i.e., that no efficient algorithm can color graphs satisfying the conditions of the theorem with \( (1-\epsilon) \Delta / \ln \sqrt{f} \) colors. More precisely, Proposition 3.1 below, proved in Appendix 6, implies that any such algorithm would entail coloring random graphs using fewer than twice as many colors as their chromatic number.

Proposition 3.1. For every \( \epsilon > 0 \) and \( d \in (d, \ln n, (\ln n)^{1/2}) \), there exist \( \Delta = \Delta(d, \epsilon) \) and \( f = f(d, \epsilon) \) such that with probability tending to 1 as \( n \to \infty \), a random graph \( G = G(n, d/n) \) satisfies the conditions of Theorem 3.5 and \( \chi(G) \geq (1/4 - \epsilon) \Delta / \ln \sqrt{f} \).
This would be a major (and unexpected) breakthrough in random graph theory, as beating this factor of two has been an elusive goal for over 40 years. Also, for sparse random graphs, this factor of two corresponds to a phase transition in the geometry of the set of colorings [1], known as the shattering threshold. In other words, our algorithm can be seen as a robust version of previously known algorithms [7] for coloring random graphs up to the shattering threshold, that applies to worst-case graphs as well.

Second, armed with Theorem 3.5 we are able to prove the following result concerning the chromatic number of general graphs, as a function of the maximum degree and the maximum number of triangles in any neighborhood:

**Theorem 3.6.** Let $G$ be a graph with maximum degree $\Delta$ in which the neighbors of every vertex span at most $\Delta^2/f$ edges. For all $\epsilon > 0$, there exist $\Delta_\epsilon, f_\epsilon$ such that if $\Delta \geq \Delta_\epsilon$ and $f \in [f_\epsilon, \Delta^2 + 1]$, then

$$\chi(G) \leq (2 + \epsilon)\Delta / \ln \sqrt{f} . \quad (7)$$

Furthermore, such a coloring can be found in polynomial time with high probability.

Theorem 3.6 improves a classical result of Alon, Krivelevich and Sudakov [11] which established (7) with an unspecified (large) constant in place of $2 + \epsilon$. Indeed, our analysis closely follows theirs. The main idea is to break down the input graph into triangle-free subgraphs, and color each one of them separately using distinct sets of colors by applying the result of Johansson [47]. Instead, we break down the graph into subgraphs with few triangles per neighborhood, and use Theorem 3.5 to color the pieces. The proof of Theorem 3.6 can be found in Appendix [E]. We note that Theorem 3.5 is essential here: even if we used Molloy’s [53] recent result in place of Johansson’s in the above scheme, the corresponding constant would still be in the thousands.

As final remark, we note that Vu [66] proved the analogue of the main result of [11] (again with a large constant) for the list chromatic number. While we don’t currently see how to sharpen Vu’s result to an analogue of Theorem 3.6 for the list chromatic number using our techniques, we note that our Theorem 3.5 improves over [66] for all $f \geq \Delta^{\frac{\ln \Delta}{\ln \ln \Delta}}$.

### 3.3 Application to Backtracking Algorithms

An important class of algorithms naturally devoid of “collateral fixes” are backtracking algorithms, as discussed in Section 2.4. In particular, consider a Constraint Satisfaction Problem (CSP) over a set of variables $V = \{v_1, v_2, \ldots, v_n\}$, each variable $v_i$ taking values in a domain $D_i$, with a set of constraints $C = \{c_1, c_2, \ldots, c_m\}$ over these variables. The backtracking algorithms we consider operate as follows. (Note that in Step 1 we can always take $\theta$ to be the distribution under which all variables are unassigned; this does not affect the convergence condition but may have a mild effect on the running time.)

| Generic Backtracking Algorithm |
|--------------------------------|
| 1: Sample a partial non-violating assignment $\sigma_0$ according to a distribution $\theta$ and set $i = 0$ |
| 2: while unassigned variables exist do |
| 3: Let $v$ be the lowest indexed unassigned variable in $\sigma_i$ |
| 4: Choose a new value for $v$ according to a state-dependent probability distribution |
| 5: if one or more constraints are violated then |
| 6: Remove the values from enough variables so that no constraint is violated |
| 7: Let $\sigma_{i+1}$ be the resulting assignment |
| 8: $i \leftarrow i + 1$ |

Let $\Omega$ be the set of partial assignments to $V$ that do not violate any constraint in $C$. For each variable $v_i \in V$, let flaw $f_i \subseteq \Omega$ comprise the partial assignments in which $v_i$ is unassigned. Clearly, each flaw $f_i$
can only be removed by addressing it, as addressing any other flaw can only unassign \(v_i\). Thus, every flaw is primary and a flawless state is a complete satisfying assignment. The fact that every flaw is primary leads to an improvement in the running time bound of the algorithm, i.e., the value of \(T_0\) in Theorem 3.3.

**Corollary 3.7.** Let \(\mathcal{I}(\theta)\) be the set comprising the sets of flaw-indices that may be present in a state selected according to \(\theta\). If every flaw is primary, then the sum over \(S \subseteq \text{Span}(\theta)\) in the definition of \(T_0\) can be restricted to \(S \in \mathcal{I}(\theta)\). In particular, if every variable is initially unassigned, this sum equals \(\sum_{i \in [m]} \log_2 \psi_i\).

We give three representative applications in Section 6. First, we develop a corollary of Theorem 3.3 that can be used to make applications of the LLL in the variable setting constructive via a backtracking algorithm, i.e., an algorithm of very different flavor from the Moser-Tardos algorithm. We note that very recently and independently, Bissacot and Doin [19] also showed that backtracking algorithms can make constructive LLL applications in the variable setting using the entropy compression method. However, their result applies only to the uniform measure and their algorithms are relatively complicated. In contrast, we show that a simple backtracking algorithm works for every product measure. Second, we show how Theorem 3.3 perfectly recovers in a black-box fashion the main result of Esperet and Parreau [30] for acyclic edge coloring. Finally, we show how our application of Theorem 5.3 to acyclic edge coloring can be adapted with minimal effort to make constructive an existential result of Bernshteyn [16] showing improved bounds for the acyclic chromatic index of graphs that do not contain any fixed arbitrary bipartite graph \(H\). Specifically, we prove the following result.

**Theorem 3.8.** Let \(G(V,E)\) be a graph with maximum degree \(\Delta\) and let \(H\) be a fixed bipartite graph. If \(G\) does not contain \(H\) as a subgraph, then there exists an acyclic edge coloring of \(G\) using at most \(3(\Delta + o(1))\) colors. Moreover, such a coloring can be found in \(O(|V| + |E|)\) time with high probability.

### 3.4 Commutative Algorithms and Distributional Properties

Besides conditions for fast convergence to flawless objects, it is natural to ask further questions about focused search algorithms, such as: “Are they parallelizable?”; “How many distinct solutions can they output?”, etc. These questions and more have been answered for the Moser-Tardos algorithm in a long series of papers [57, 38, 43, 50, 22, 23, 37, 27, 2]. As a prominent example, the result of Haeupler, Saha and Srinivasan [38] shows that the Moser-Tardos algorithm, in a certain sense, approximates well the LLL-distribution, i.e., the distribution obtained by conditioning on avoiding all bad events.

Harvey and Vondrák [44] showed that these distributional results are unlikely to transfer to the more general algorithmic LLL settings of [4, 44, 5], because the so-called Witness Tree Lemma—the key technical ingredient for analyzing the Moser-Tardos algorithm—can fail to hold in such settings. In [40], Iliopoulos established the Witness Tree Lemma for algorithms satisfying Kolmogorov’s notion of “commutativity” [52] and showed how it can be used to establish many distributional properties of such algorithms. Kolmogorov’s notion of commutativity requires that for every \(i \sim j \in [m]\), every sequence of state transitions of the form \(\sigma_1 \xrightarrow{1} \sigma_2 \xrightarrow{2} \sigma_3\) can be mapped to a distinct sequence of state transitions of the form \(\sigma_1 \xrightarrow{1} \sigma_2' \xrightarrow{2} \sigma_3\), so that \(\rho_i(\sigma_1, \sigma_2) \rho_j(\sigma_2, \sigma_3) = \rho_j(\sigma_1, \sigma_2') \rho_i(\sigma_2', \sigma_3) > 0\).

Our matrix framework allows us to introduce a more natural notion of algorithmic commutativity, essentially matrix commutativity, that is also more general than the notion of [52]. For \(i \in [m]\), let \(A_i\) denote the \(|\Omega| \times |\Omega|\) matrix where \(A_i[\sigma, \sigma'] = \rho_i(\sigma, \sigma')\) for \(\sigma \in f_i\), and 0 otherwise. Recall the definition of \(\Gamma(i)\).

**Definition 3.9.** An algorithm is commutative with respect to a symmetric binary relation \(\sim\) if

(a) \(A_i A_j = A_j A_i\), for every \(i, j \in [m]\) such that \(i \sim j\).

(b) \(\Gamma(i) \subseteq \{j : i \sim j\}\).
Remark 3.6. In most applications $A_i A_j \neq A_j A_i$ when $i \in \Gamma(j)$, in which case (a) implies (b).

Under this new notion, we recover all the results of [52, 46] with much simpler proofs, at the mild cost of restricting the family of flaw choice strategies to canonical ones, per Definition 3.10 below. (In [52, 46] the flaw choice strategy can be arbitrary.) Note that in the commutative setting, canonical flaw choice strategies suffice to capture the optimal convergence results, so that the restriction to such strategies is indeed mild.

Definition 3.10. Fix an arbitrary sequence of (possibly stochastic) functions $(s_i)_{i \geq 1} : \Omega \to [m]$, each $s_i$ mapping $\sigma \in \Omega$ to an element of $U(\sigma)$. A flaw choice strategy is canonical if the flaw addressed in the $i$-th step is $s_i(\sigma_i)$, where $\sigma_i \in \Omega$ is the state after $i$ steps.

In particular, we establish the Witness Tree Lemma, from which all the other results follow. (In [52, 46] the flaw choice strategy can be arbitrary.) Note that in the commutative setting, canonical flaw choice strategies restricting the family of flaw choice strategies to canonical ones.

Remark 3.6. In most applications $A_i A_j \neq A_j A_i$ when $i \in \Gamma(j)$, in which case (a) implies (b).

Throughout this section we use standard facts about operator norms, summarized briefly in Appendix A.

Theorem 3.11 (Informal Statement). The Witness Tree Lemma holds for commutative algorithms that follow canonical flaw choice strategies.

4 Proof of Main Theorem

In Sections 4.1 and 4.2 we present the proof of our main result, the new algorithmic LLL condition in Theorem 3.3. In Section 4.3 we show how to extend the theorem to allow flaw choice strategies other than following a fixed permutation over flaws.

Throughout this section we use standard facts about operator norms, summarized briefly in Appendix A.

4.1 Tracking the Set of Current Flaws

We say that a trajectory $\Sigma = (\sigma_1, \sigma_2, \ldots, \sigma_t)$ followed by the algorithm is a bad $t$-trajectory if every state $\sigma_i$, $i \in [t+1]$, is flawed. Thus, our goal is to bound the probability that the algorithm follows a bad $t$-trajectory.

Given a bad trajectory, intuitively, we track the flaws introduced into the state in each step, where a flaw is said to “introduce itself” whenever addressing it fails to remove it. Of the flaws introduced in each step, we disregard those that later get eradicated collaterally, i.e., by an action addressing some other flaw. The rest form the “witness sequence” of the trajectory, i.e., a sequence of sets of flaws.

Fix any permutation $\pi$ on $[m]$. For any $S \subseteq [m]$, let $\pi(S) = \min_{j \in S} \pi(j)$, i.e., the lowest index in $S$ according to $\pi$. Recalling that $U(\sigma)$ is the set of indices of flaws present in $\sigma$, in the following we assume that the index of the flaw addressed in state $\sigma$ is $\pi(U(\sigma))$, which we sometimes abbreviate as $\pi(\sigma)$. Also, to lighten notation, we will denote $A \setminus \{\pi(B)\}$ by $A - \pi(B)$.

Definition 4.1. Let $\Sigma = (\sigma_1, \sigma_2, \ldots, \sigma_t)$ be any bad $t$-trajectory. Let $B_0 = U(\sigma_1)$. For $1 \leq i \leq t$, let

$$B_i = U(\sigma_{i+1}) \setminus [U(\sigma_i) - \pi(\sigma_i)],$$

i.e., $B_i$ comprises the indices of the flaws introduced in the $i$-th step. For $0 \leq i \leq t$, let

$$C_i = \{k \in B_i \mid \exists j \in [i + 1, t] : k \notin U(\sigma_{j+1}) \wedge \forall \ell \in [i + 1, j] : k \neq \pi(\sigma_{\ell})\},$$

i.e., $C_i$ comprises the indices of the flaws introduced in the $i$-th step that get eradicated collaterally. The witness sequence of bad $t$-trajectory $\Sigma$ is the sequence of sets

$$w(\Sigma) = (B_0 \setminus C_0, B_1 \setminus C_1, \ldots, B_t \setminus C_t).$$
A crucial feature of witness sequences is that they allow us to recover the sequence of flaws addressed.

**Definition 4.2.** Given an arbitrary sequence \(S_0, \ldots, S_t\), let \(S^*_1 = S_0\), while for \(1 \leq i \leq t\), let

\[
S^*_{i+1} = \begin{cases} 
[S^*_i - \pi(S^*_i)] \cup S_i & \text{if } S^*_i \neq \emptyset, \\
\emptyset & \text{otherwise}.
\end{cases}
\]

If \(S^*_i \neq \emptyset\) for all \(1 \leq i \leq t\), then we say that \((S_i)_{i=0}^t\) is plausible and write \(\pi(S^*_i) = (i)\).

**Lemma 4.3.** If \(\Sigma = (\sigma_1, \sigma_2, \ldots, \sigma_{t+1})\) is any bad t-trajectory, then \(w(\Sigma) = (S_0, \ldots, S_t)\) is plausible, \(\pi(\sigma_i) = \pi(S^*_i) = (i)\) for all \(1 \leq i \leq t\), and for every flaw index \(z \in [m]\), the number of times \(z\) occurs in the multiset \(\bigcup_{i=0}^t S_i\) minus the number of times it occurs in the multiset \(\bigcup_{i=1}^t (i)\) equals \(1_{z \in S^*_{t+1}}\).

**Proof.** Recall that \(S_i = B_i \setminus C_i\). For \(1 \leq i \leq t+1\), \(L_i\) comprise the elements of \(U(\sigma_i)\) eradicated collaterally during the \(i\)-th step and let \(H_i\) comprise the elements of \(U(\sigma_i)\) eradicated collaterally during any step \(j \geq i\). Observe that \(H_{i+1} = (H_i \setminus L_i) \cup C_i\). We will prove, by induction, that for all \(1 \leq i \leq t+1\),

\[
S^*_i \subseteq U(\sigma_i) \quad U(\sigma_i) \setminus S^*_i = H_i.
\]  

(8)  

(9)

Observe that if (8), (9) hold for a given \(i\), then \(\pi(\sigma_i) = \pi(S^*_i)\), since \(\pi(\sigma_i) \notin H_i\) by the definition of \(H_i\), and \(\pi(A) = \pi(A \setminus B)\) whenever \(\pi(A) \notin B\). Moreover, \(S^*_i \neq \emptyset\), because otherwise \(U(\sigma_i) = H_i\), an impossibility. To complete the proof it suffices to note that for any \(z \in [m]\), the difference in question equals \(1_{z \in U(\sigma_{i+1})} - 1_{z \in U(\sigma_{i+1})}\times S^*_{i+1}\) since, by definition, \(H_{t+1} = \emptyset\). The inductive proof is as follows.

For \(i = 1\), (8), (9) hold since \(S^*_1 = B_0 \setminus C_0\), while \(U(\sigma_1) = B_0\). If (8), (9) hold for some \(i \geq 1\), then \(S^*_i = [S^*_i - \pi(\sigma_i)] \cup S_i\) while, by definition, \(U(\sigma_{i+1}) = [U(\sigma_i) - \pi(\sigma_i) \setminus L_i] \cup B_i\). Thus, the fact that \(S^*_i \subseteq U(\sigma_i)\) trivially implies \(S^*_{i+1} \subseteq U(\sigma_{i+1})\), while

\[
U(\sigma_{i+1}) \setminus S^*_{i+1} = (U(\sigma_i) \setminus S^*_i \setminus L_i) \cup (B_i \setminus S_i) = (H_i \setminus L_i) \cup C_i = H_{i+1}.
\]

\(\square\)

The first step in our proof of Theorem 3.3 is to give an upper bound on the probability that a given witness sequence occurs in terms of the charges \(\gamma^S_i\) defined in Section 3.1.

**Lemma 4.4.** Fix any integer \(t \geq 0\) and let \(\Sigma\) be the random variable \((\sigma_1, \ldots, \sigma_{t+1})\). For any invertible matrix \(M\) such that \(\sum_{\tau \in \Omega} \|Me_\tau\| = 1\) and any plausible sequence \(\phi = (S_0, \ldots, S_t)\),

\[
\Pr[w(\Sigma) = \phi] \leq \|\theta^T M^{-1}\|_s \prod_{i=1}^t \gamma^S_i.
\]

(10)

**Proof.** By Definition 4.1 and Lemma 4.3 a necessary condition for \(w(\Sigma) = \phi\) to occur is that \((i) \in U(\sigma_i)\) and \(S_i \subseteq B_i\), for every \(1 \leq i \leq t\).

Recall that for any \(S \subseteq [m]\), we denote by \(S^P\) and \(S^N\) the subsets of \(S\) that correspond to primary and non-primary flaws, respectively. By Definition 4.1 and Lemma 4.3 a necessary condition for \(w(\Sigma) = \phi\) to occur is that \((i) \in U(\sigma_i)\) and \(S_i \subseteq B_i\), for every \(1 \leq i \leq t\). Moreover, since primary flaws are never eradicated collaterally, i.e., \(C^P_i = \emptyset\) always, it must also be that \(S^P_i = B^P_i\) for \(1 \leq i \leq t\). Fix any state \(\tau \in \Omega\). If \(\theta^T \in [0,1[^{|\Omega|}\) is the row vector expressing the probability distribution of the initial state \(\sigma_1\), then
the probability that \((1) \in U(\sigma_1) \land S^P = B^P_i(\Sigma) \land S^N = B^N_i(\Sigma) \land \sigma_2 = \tau\) equals the \(\tau\)-column (coordinate) of the row-vector \(\theta^T A^S_i(1)\). More generally, if \(e_\sigma\) is the indicator vector of state \(\sigma\), we see that for any \(t \geq 1\),

\[
\Pr \left[ \bigwedge_{i=1}^{t} ((i) \in U(\sigma_i)) \land \bigwedge_{i=1}^{t} (S^P_i = B^P_i) \land \bigwedge_{i=1}^{t} (S^N_i \subseteq B^N_i) \land \sigma_{t+1} = \tau \right] = \theta^T \prod_{i=1}^{t} A^S_i e_\tau .
\] (11)

Consider now any vector norm \(\| \cdot \|\) and the corresponding operator norm. By (36),

\[
\theta^T \prod_{i=1}^{t} A^S_i e_\tau = \theta^T M^{-1} \left( \prod_{i=1}^{t} MA^S_i M^{-1} \right) M e_\tau \leq \left\| \theta^T M^{-1} \left( \prod_{i=1}^{t} MA^S_i M^{-1} \right) \right\| \| M e_\tau \| .
\] (12)

Summing (12) over all \(\tau \in \Omega\) and restricting to matrices \(M\) for which \(\sum_{\tau \in \Omega} \| M e_\tau \| = 1\) we conclude that

\[
\Pr[w(\Sigma) = \phi] = \sum_{\tau \in \Omega} \Pr[w(\Sigma) = \phi \land \sigma_{t+1} = \tau] \leq \left\| \theta^T M^{-1} \prod_{i=1}^{t} MA^S_i M^{-1} \right\| \| M e_\tau \| .
\] (13)

Applying (38) and then (37) to (13) and recalling the definition of \(\gamma^S_i\) we conclude that

\[
\Pr[w(\Sigma) = \phi] \leq \| \theta^T M^{-1} \|_* \prod_{i=1}^{t} \| MA^S_i M^{-1} \| = \| \theta^T M^{-1} \|_* \prod_{i=1}^{t} \gamma^S_i ,
\]

as claimed. \(\square\)

Let \(F_t = \{ w(\Sigma) : \Sigma\text{ is a bad }t\text{-trajectory of the algorithm} \}\). Since \(F_t\) contains only plausible sequences, an immediate corollary of Lemma 4.4 is a bound on the probability that the algorithm fails in \(t\) steps.

**Corollary 4.5.** The probability that the algorithm fails to reach a flawless state within \(t\) steps is at most

\[
\| \theta^T M^{-1} \|_* \sum_{\phi \in F_t} \prod_{i=1}^{t} \gamma^S_i .
\] (14)

Thus, to complete the proof of Theorem 3.3 we are left with the task of bounding the sum in (14).

### 4.2 Bounding the Sum

Given \(\psi_1, \ldots, \psi_m > 0\) and \(S \subseteq [m]\), let \(\Psi(S) = \prod_{j \in S} \psi_j\), with \(\Psi(\emptyset) = 1\). For each \(i \in [m]\), let

\[
\zeta_i = \frac{1}{\psi_i} \sum_{S \subseteq [m]} \gamma_i^S \Psi(S) .
\]

Finally, for each \(i \in [m]\) consider the probability distribution on \(2^{[m]}\) assigning to each \(S \subseteq [m]\) probability

\[
p(i, S) = \frac{\gamma_i^S \Psi(S)}{\sum_{S \subseteq [m]} \gamma_i^S \Psi(S)} = \frac{\gamma_i^S \Psi(S)}{\zeta_i \psi_i} .
\]

For any \(S_0 \subseteq [m]\), let \(F_t(S_0)\) comprise the witness sequences in \(F_t\) whose first set is \(S_0\). Consider the probability distribution on sequences of subsets of \([m]\) generated as follows: \(R_1 = S_0\); for \(i \geq 1\), if \(R_i \neq \emptyset\), then \(R_{i+1} = (R_i - \pi(R_i)) \cup S_i\), where \(\Pr[S_i = S] = p(\pi(R_i), S)\), for any \(S \subseteq [m]\). Under this
distribution, by Lemma 4.3 each \( \phi = (S_0, \ldots, S_t) \in \mathcal{F}_t(S_0) \) receives probability \( p_\phi = \prod_{i=1}^t p((i), S_i) \), while \( \sum_{\phi \in \mathcal{F}_t(S_0)} p_\phi \leq 1 \). At the same time, by the last claim in Lemma 4.3

\[
p_\phi = \prod_{i=1}^t p((i), S_i) = \left( \prod_{i=1}^t p((i), S_i) \frac{\psi(S_i)}{\Psi(S_i)} \right) \frac{\Psi(S_{i+1})}{\Psi(S_i)} \prod_{i=1}^t \frac{\gamma_i^{S_i}}{\zeta_i}.
\]  

Combining (15) with the fact \( \sum_{\phi \in \mathcal{F}_t(S_0)} p_\phi \leq 1 \) it follows that

\[
\sum_{\phi \in \mathcal{F}_t(S_0)} \prod_{i=1}^t \frac{\gamma_i^{S_i}}{\zeta_i} \leq \max_{S \subseteq \{m\}} \frac{\Psi(S)}{\Psi(S)}.
\]  

Let \( \zeta = \max_i \{ \zeta_i \} \). Then, summing equation (16) over all possible sets \( S_0 \) yields

\[
\sum_{\phi \in \mathcal{F}_t} \prod_{i=1}^t \frac{\gamma_i^{S_i}}{\zeta_i} \leq \zeta^t \sum_{S_0 \subseteq \text{Span}(\theta)} \sum_{\phi \in \mathcal{F}_t(S_0)} \prod_{i=1}^t \frac{\gamma_i^{S_i}}{\zeta_i} \leq \max_{S \subseteq \{m\}} \sum_{S_0 \subseteq \text{Span}(\theta)} \frac{\Psi(S)}{\Psi(S)}.
\]  

Proof of Theorem 3.3 Combining (17) with Corollary 4.5 we see that the binary logarithm of the probability that the algorithm does not encounter a flawless state within \( t \) steps is at most \( t \log_2 \zeta + T_0 \), where

\[
T_0 = \log_2 \|\theta^\top M^{-1}\|_\psi + \log_2 \left( \sum_{S \subseteq \text{Span}(\theta)} \Psi(S) \right) + \log_2 \left( \max_{S \subseteq \{m\}} \frac{1}{\Psi(S)} \right).
\]

Therefore, if \( t = (T_0 + s)/\log_2(1/\zeta) \leq (T_0 + s)/\delta \), the probability that the algorithm does not reach a flawless state within \( t \) steps is at most \( 2^{-s} \). \( \square \)

We conclude by proving Corollary 3.7 concerning backtracking algorithms, stated in Section 3.3.

Proof of Corollary 3.7 When every flaw is primary, the only equivalence classes of \( \mathcal{F}_t \) that contribute to the sum in (17) are those for which \( S_0 \in \mathcal{I}(\theta) \). Thus, for backtracking algorithms the sum over \( S \subseteq \text{Span}(\theta) \) in the definition of \( T_0 \) can be restricted to \( S \subseteq \mathcal{I}(\theta) \). If, in such algorithms, \( \theta \) is such that every variable is initially unassigned, then \( \mathcal{I}(\theta) = \{ [m] \} \), in which case the sum over \( \mathcal{I}(\theta) \) equals \( \sum_{i \in [m]} \log_2 \psi_i \). \( \square \)

4.3 Other flaw choice strategies

The only place where we used the fact that the flaw choice is based on a fixed permutation was to assert, in Lemma 4.3, that the witness sequence of a trajectory determines the sequence of addressed flaws. Thus, our analysis is in fact valid for every flaw choice strategy that shares this property.

A first example of such a strategy is “pick a random occurring flaw and address it”. To implement this, we can fix a priori an infinite sequence of uniformly random permutations \( \pi_1, \pi_2, \ldots \) and at the \( i \)-th step address the lowest indexed flaw present according to \( \pi_i \). It is straightforward to see that Lemma 4.3 still holds if we replace \( \pi \) with \( \pi_i \) therein and in Definition 4.2.

As a second example, consider the following recursive way to choose which flaw to address at each step (which makes the algorithm non-Markovian). The algorithm now maintains a stack. The flaws present in \( \sigma_1 \), ordered according to some permutation \( \pi \), comprise the initial stack content. The algorithm starts by addressing the flaw at the top of the stack, i.e., \( \pi(\sigma_1) \), as before. Now, though, any flaws introduced in the \( i \)-th step, i.e., the elements of \( B_i \), go on the top of the stack (ordered by \( \pi \)), while all eradicated flaws are removed from the stack. The algorithm terminates when the stack empties. It is not hard to see that, by taking \( S_0 \) to be the initial stack content, popping the flaw at the top of the stack at each step, and adding \( S_i \) to the top of the stack (ordered by \( \pi \)), the sequence of popped flaws is the sequence of addressed flaws.
5 Graph Coloring Proofs

5.1 The Algorithm

To prove Theorem 3.5 we will generalize the algorithm of Molloy [53] for coloring triangle-free graphs. The main issue we have to address is that in the presence of triangles, the natural generalization of Molloy’s algorithm introduces monochromatic edges when the neighborhood of a vertex is recolored. As a result, the existing analysis fails completely even if each vertex participates in just one triangle. To get around this problem, we introduce backtracking steps into the algorithm, whose analysis is enabled by our new convergence condition, Theorem 3.3.

For each vertex $v \in V$, let $N_v$ denote the neighbors of $v$ and let $E_v = \{\{u_1, u_2\} : u_1, u_2 \in N_v\}$ denote the edges spanned by them. Recall that the color-list of $v$ is denoted by $L_v$. It will be convenient to treat $\text{Blank}$ also as a color. Indeed, the initial distribution $\theta$ of our algorithm assigns all its probability mass to the state where every vertex is colored $\text{Blank}$. Whenever assigning a color to a vertex creates monochromatic edges, the algorithm will immediately uncolor enough vertices so that no monochromatic edge remains. Edges with two $\text{Blank}$ endpoints are not considered monochromatic. To uncolor a vertex $v$, the algorithm picks a monochromatic edge $e$ incident to $v$ and assigns $e$ to $v$ instead of a color, thus also creating a record of the reason for the uncoloring. Thus, 

$$\Omega \subseteq \prod_{v \in V} \{L_v \cup \{\text{Blank}\} \cup E_v\}.$$ 

Let $L = (1 + \epsilon)\frac{\Delta}{\ln f} f^{-\frac{1}{2r+2 \epsilon}}$ and assume $\Delta$ is sufficiently large so that $L \geq 10$.

5.1.1 The Flaws

We let $L_v(\sigma) \subseteq (L_v \cup \{\text{Blank}\})$ be the set of colors we can assign to $v$ in state $\sigma$ without creating any monochromatic edge. We call these the available colors for $v$ in $\sigma$ and note that $\text{Blank}$ is always available. For each $v \in V$, we define a flaw expressing that there are “too few available colors for $v$,” namely 

$$B_v = \{\sigma \in \Omega : |L_v(\sigma)| < L\}.$$

For each color $c$ other than $\text{Blank}$, let $T_{v,c}(\sigma)$ be the set of $\text{Blank}$ neighbors of $v$ for which $c$ is available in $\sigma$, i.e., the vertices that may “compete” with $v$ for color $c$. For each $v \in V$, we define a flaw expressing that there is “too much competition for $v$’s available (real) colors,” namely 

$$Z_v = \left\{\sigma \in \Omega : \sum_{c \in L_v(\sigma) \setminus \text{Blank}} |T_{v,c}(\sigma)| > \frac{L}{10}|L_v(\sigma)|\right\}.$$

Finally, for each $v \in V$ and $e \in E$ we define a flaw expressing that $v$ is uncolored (because of $e$), namely 

$$f_v^e = \{\sigma \in \Omega : \sigma(v) = e\}.$$

Let $F_v = B_v \cup Z_v \cup_{e \in E} f_v^e$ and let $\Omega^+ = \Omega - \cup_{v \in V} F_v$.

Lemma 5.1 (The Second Phase [53]). Given $\sigma \in \Omega^+$, a complete list-coloring of $G$ can be found efficiently.

The proof of Lemma 5.1 is a fairly standard application of the (algorithmic) LLL, showing that $\sigma$ can be extended to a complete list-coloring by coloring all $\text{Blank}$ vertices with actual colors. Thus, the heart of the matter is reaching a state i.e., (a partial coloring) not suffering from any of the flaws specified above.
5.1.2 Flaw Choice

The algorithm can use any \(\pi\)-strategy in which every \(B\)-flaw has priority over every \(f\)-flaw.

5.1.3 The Actions

To address \(f^v_\sigma\) at \(\sigma\), i.e., to color \(v\), the algorithm simply chooses a color from \(L_v(\sigma)\) uniformly and assigns it to \(v\). The fact that \(B\)-flaws have higher priority than \(f\)-flaws implies that there are always at least \(L\) choices.

Addressing \(B\)- and \(Z\)-flaws is significantly more sophisticated. For each vertex \(v\), for each vertex \(u \in N_v\), let \(R^v_u(\sigma) \supseteq L_u(\sigma)\) comprise those colors having the property that assigning them to \(u\) in state \(\sigma\) creates no monochromatic edge except, perhaps, in \([u, v] \in E\). To address either \(B_v\) or \(Z_v\) in \(\sigma\), the algorithm performs the following, i.e., the set of actions and the distribution on them as induced by the following procedure.

\[
\begin{align*}
1: & \textbf{procedure RECOLOR}(v, \sigma) \\
2: & \quad \text{Assign to each colored vertex } u \text{ in } N_v \text{ a uniformly random color from } R^v_u(\sigma) \\
3: & \quad \textbf{while} \text{ monochromatic edges exist} \textbf{do} \\
4: & \quad \quad \text{Let } u \text{ be the lowest indexed vertex participating in a monochromatic edge} \\
5: & \quad \quad \text{Let } e \text{ be the lowest indexed monochromatic edge with } u \text{ as an endpoint} \\
6: & \quad \quad \text{Uncolor } u \text{ by assigning } e \text{ to } u
\end{align*}
\]

**Lemma 5.2.** Let \(S'(v, \sigma)\) be the set of colorings that can be reached at the end of Step 2 of RECOLOR\((v, \sigma)\) and let \(S''(v, \sigma)\) be the set of possible final colorings. Then \(|S'(v, \sigma)| = |S''(v, \sigma)|\).

**Proof.** Since Steps 4–6 are deterministic, \(|S''(v, \sigma)| \leq |S'(v, \sigma)|\). To prove that \(|S''(v, \sigma)| \geq |S'(v, \sigma)|\), we will prove that if \(u \in N_v\) has distinct colors in \(\sigma'_1, \sigma'_2 \in S'\), then there exists \(z \in V\) such that \(\sigma''_1(z) \neq \sigma''_2(z)\).

Imagine that in Step 6 we also oriented \(e\) to point away from \(u\). Then, in the resulting partial orientation, every vertex would have outdegree at most 1 and there would be no directed cycles. Consider the (potentially empty) oriented paths starting at \(u\) in \(\sigma''_1\) and \(\sigma''_2\) and let \(z\) be their last common vertex. If \(z\) is uncolored, then \(\sigma''_1(z) = e_1\) and \(\sigma''_2(z) = e_2\), where \(e_1 \neq e_2\); if \(z\) is colored, then \(\sigma''_1(z) = \sigma''_2(z)\).

5.2 Proving Termination

Let \(D_v\) be the set of vertices at distance 1 or 2 from \(v\) and let

\[
S_v = \{B_u\}_{u \in D_v} \cup \{Z_u\}_{u \in D_v} \cup \{f^u_w\}_{u,w \in N_v}.
\]

To lighten notation, in the following we write \(\gamma^S(f)\) instead of \(\gamma^S(f^u_w)\). Let \(q = (1 + \epsilon)\frac{\Delta}{\ln \sqrt{f}} \geq 1\).

**Lemma 5.3.** For every vertex \(v \in V\) and edge \(e \in E\),

(a) if \(S \not\subseteq S_v\), then \(\gamma^S(B_v) = \gamma^S(Z_v) = \gamma^S(f_e^v) = 0\);

(b) if \(S \supseteq \{f^u_w\}_{u \in D_v} \cup \{f^{u_1}_{w_2}\}\), then \(\gamma^S(B_v) = \gamma^S(Z_v) = \gamma^S(f_e^v) = 0\).

(c) \(\max_{S \subseteq F} \gamma^S(f_e^v) \leq \frac{1}{S} =: \gamma(f_e^v)\);

(d) \(\max_{S \subseteq F} \gamma^S(B_v) \leq 2e^{-\frac{1}{S}} : = \gamma(B_v)\);

(e) \(\max_{S \subseteq F} \gamma^S(Z_v) \leq 3qe^{-\frac{1}{S}} =: \gamma(Z_v)\).
We note that while we give uniform bounds on the charges corresponding to each flaw, the analysis of our algorithm cannot be captured by (5). This is crucial since it allows us to consider refined charges per our discussion in Section 2.4.

Before we give the proof, we first use Lemma 5.3 to derive Theorem 5.3.

Proof of Theorem 5.3. We will apply Theorem 3.3 with \( M = I/\Omega \) and the \( \| \cdot \|_1 \) norm. For every flaw \( f \in F \), we will take \( \psi_f = \gamma(f)\psi \), where \( \psi > 0 \) will be chosen later.

For any vertex \( v \in V \), flaw \( f \in \{B_v,Z_v,f_v^c\} \), and set of flaws \( S \subseteq F \), Lemma 5.3 implies that \( \gamma^S(f) = 0 \) unless all \( B \) - and \( Z \) -flaws in \( S \) correspond to vertices in \( D_v \), per part (a), and every edge \( e \in E_v \) contributes at most one flaw to \( S \), per part (b). Therefore, for \( f \in \{B_v,Z_v,f_v^c\} \),

\[
\frac{1}{\psi_f} \sum_{S \subseteq F} \gamma^S(f) \prod_{g \in S} \psi_g \leq \frac{1}{\psi} \prod_{u \in D_v} (1+\gamma(B_u)\psi)(1+\gamma(Z_u)\psi) \prod_{e=(u_1,u_2) \in E_v} (1+\gamma(f^e_{u_1})\psi+\gamma(f^e_{u_2})\psi) .
\]

(18)

To bound the right hand side of (18) we use parts (c)–(e) of Lemma 5.3 along with the facts \(|D_v| \leq \Delta^2 + 1 \) and \(|E_v| \leq \Delta^2/f \) to derive (19) below. To derive (20), we use the fact that \( 2e^{-\frac{\Delta}{2\psi}} \leq 3qe^{-\frac{\Delta}{6\psi}} \), since \( q \geq 1 \), and that \( 1 + x \leq e^x \) for all \( x \). Thus, for \( f \in \{B_v,Z_v,f_v^c\} \), we conclude

\[
\frac{1}{\psi} \sum_{S \subseteq F} \gamma^S(f) \prod_{g \in S} \psi_g \leq \frac{1}{\psi} \left( 1 + 2e^{-\frac{\Delta}{2\psi}} \right)^{\Delta^2+1} \left( 1 + 3qe^{-\frac{\Delta}{6\psi}} \right)^{\Delta^2+1} \left( 1 + 2\psi \right)^{\frac{\Delta^2}{f}} \left(1 + \frac{2\psi}{L} \right)
\]

\[
\leq \frac{1}{\psi} \exp \left( \frac{2\psi \Delta^2}{fL} + 6qe^{-\frac{\Delta}{6\psi}} \psi(\Delta^2 + 1) \right) := \frac{1}{\psi} \exp(Q) .
\]

(19)

Setting \( \psi = (1+\epsilon) \), we see that the right hand side of (20) is strictly less than 1 for all \( \Delta \geq \Delta_c \), since \( Q \overset{\Delta \to \infty}{\longrightarrow} 0 \) for all \( f \in [\Delta - \frac{\Delta^2}{2\psi}]^2, \Delta^2 + 1 \). To see this last claim, recall that \( L = (1+\epsilon)\frac{\Delta}{\ln f} f^{-\frac{1}{1+2\psi}} \) and \( q = (1+\epsilon)\frac{\Delta}{\ln f} \), and note that \( \ln f < 3 \ln \Delta \) and \( f^{\frac{1}{1+2\psi}} \geq \Delta(\ln \Delta)^{\frac{2\psi + 1}{1+2\psi}} \). Thus,

\[
\frac{2\psi \Delta^2}{fL} = \frac{2\Delta^2}{f^\frac{1}{1+2\psi}} = \frac{2\Delta \ln f}{f^\frac{1}{1+2\psi}} \leq \frac{2 \ln f}{(\ln \Delta)^{\frac{1+2\psi}{1+2\psi}}} \leq \frac{6 \ln \Delta}{(\ln \Delta)^{\frac{2\psi + 1}{1+2\psi}}} = \frac{6 e^{-\frac{\Delta}{6\psi}} \psi(\Delta^2 + 1)}{\ln \Delta} \overset{\Delta \to \infty}{\longrightarrow} 0 ,
\]

(21)

while the facts \( L = \Omega(\Delta^{-\frac{1}{1+2\psi}}) \) and \( q \leq (1+\epsilon)\Delta \) imply that \( 6qe^{-\frac{\Delta}{6\psi}} \psi(\Delta^2 + 1) \overset{\Delta \to \infty}{\longrightarrow} 0 \).

\( \square \)

5.2.1 Proof of Lemma 5.3

Proof of part (a). Addressing \( B_v \) or \( Z_v \) by executing RECOLOR \((v, \cdot)\) only changes the color of vertices in \( N_v \), with any resulting uncolorings being due to edges in \( E_v \). Thus, only flaws in \( S_v \) may be introduced. Addressing \( f_v^c \), by coloring \( v \), trivially, can only introduce flaws \( B_u, Z_u \), where \( u \in N_v \).

\( \square \)

Proof of part (b). Since addressing an \( f \)-flaw never introduces another \( f \)-flaw, we only need to discuss procedure RECOLOR. Therein, vertices are uncolored serially in time, so that any time a vertex \( w \) is uncolored there exists, at the time of \( w \)'s uncoloring, a monochromatic edge \( e = \{w,u\} \). Therefore, an edge \( e = \{u_1,u_2\} \) can never be the reason for the uncoloring of both its endpoints, i.e., \( f_v^c_{u_1} \cap f_v^c_{u_2} = \emptyset \).

\( \square \)

Proof of part (c). If addressing \( f_v^c \) results in \( \sigma' \), then the previous state \( \sigma \) must be the mutation of \( \sigma' \) that results by assigning \( e \) to \( v \). Since \( \pi(\sigma) = f_v^c \) implies \( \sigma \notin B_v \), it follows that \( |L_v(\sigma)| \geq L \). Since colors are chosen uniformly from \( L_v(\sigma) \), it follows that \( e(f_v^c \leq 1/L) \).

\( \square \)
Proof of parts (\(d\)) and (\(e\)). Observe that every flaw corresponding to an uncolored vertex is primary since procedure \textsc{Recolor} never colors an uncolored vertex and addressing \(f_v^c\) only colors \(v\). Thus, when computing \(\gamma^S(f)\), for \(f \in \{B_v, Z_v\}\) and \(S \subseteq F\), we can restrict to pairs \((\sigma, \sigma')\) such that the set of uncolored vertices in \(\sigma'\) is exactly the union of the set of uncolored vertices in \(\sigma\) and the set \(\{u \in N_v : f_u^c \in S\}\). Fixing \(f \in \{B_v, Z_v\}\), \(S \subseteq F\), and \(\sigma'\), let us denote by \(\text{Ins}(f, \sigma')\) the candidate set of originating states and by \(U_S(f, \sigma')\) their common set of uncolored vertices. Then, for any \(f \in \{B_v, Z_v\}\) and any \(S \subseteq F\),

\[
\gamma^S(f) = \max_{\sigma' \in \Omega} \sum_{\sigma \in \text{Ins}(f, \sigma')} \rho_f(\sigma, \sigma') .
\]

To bound \(\rho_f(\sigma, \sigma')\) in (22), we recall that \textsc{Recolor} assigns to each uncolored vertex \(u \in N_v\) a random color from \(R_u^v(\sigma)\) and invoke Lemma 5.2 to derive the first equality in (23). For the second equality we observe that for every \(u \in N_v\), the set \(R_u^v\) is determined by the colors of the vertices in \(V \setminus N_v\). Since \textsc{Recolor} only changes the color of vertices in \(N_v\), it follows that \(R_u^v(\sigma) = R_u^v(\sigma')\), yielding

\[
\rho_f(\sigma, \sigma') = \frac{1}{\prod_{u \in N_v \setminus U_S(f, \sigma')} |R_u^v(\sigma)|} = \frac{1}{\prod_{u \in N_v \setminus U_S(f, \sigma')} |R_u^v(\sigma')|} := \frac{1}{\Lambda_S(f, \sigma')} .
\]

Next we bound \(|\text{Ins}(f, \sigma')|\), as follows. First we observe that if \(\sigma \in \text{Ins}(f, \sigma')\), then \(\sigma(u) \neq \sigma'(u)\) implies \(u \in N_v \setminus U_S(f, \sigma')\) and, therefore, \(\sigma(u) \in L_u(\sigma) \subseteq R_u^v(\sigma)\). Thus, the set of \(\sigma'\)-mutations that result by recoloring each vertex in \(N_v \setminus U_S(f, \sigma')\) with a color from \(R_u^v(\sigma')\) so that the resulting state belongs in \(f\) is a superset of \(\text{Ins}(f, \sigma')\). Denoting this last set by \(\text{Viol}(f, \sigma')\), we conclude that

\[
\gamma^S(f) = \max_{\sigma' \in \Omega} |\text{Ins}(f, \sigma')| \leq \max_{\sigma' \in \Omega} |\text{Viol}(f, \sigma')| = \max_{\sigma' \in \Omega} \text{Pr}[\text{Recolor}(v, \sigma') \in f] ,
\]

where for the last equality we use the definition of \(\Lambda_S(f, \sigma')\).

Remark 5.1. We note that expressing the sum of the transition probabilities into a state in terms of a random experiment as we do in (24) was the key technical idea of [53] in order to apply the entropy compression method. It is also the one that breaks down if we allow our algorithm to go through improper colorings.

To conclude the proof of Lemma 5.3, we prove the following in Appendix D.

Lemma 5.4. For each vertex \(v\) and \(\sigma \in \Omega:\)

(a) \(\text{Pr}[\text{Recolor}(v, \sigma') \in B_v] \leq 2e^{-\frac{L}{60}}\).

(b) \(\text{Pr}[\text{Recolor}(v, \sigma') \in Z_v] \leq 3e^{-\frac{L}{60}}\).

\(\Box\)

6 Applications to Backtracking Algorithms

In this section we present two applications of our main theorem to analyze backtracking search algorithms. First, we prove a useful corollary of Theorem 3.3 that holds in the so-called variable setting. Second, we analyze a backtracking algorithm for acyclic edge coloring that lies outside the variable setting. We emphasize that these analyses follow very easily from our framework.
6.1 The Variable Setting

In this section we show how we can use Theorem 3.3 to employ backtracking algorithms in order to capture applications in the variable setting, i.e., the setting considered by Moser and Tardos. In particular, we consider a product measure over variables $V$ and define a bad event for each constraint $c \in C$ being violated and show the following corollary of Theorem 3.3.

**Theorem 6.1.** Let $P$ be any product measure over a set of variables $V$ and let $A_c$ be the event that constraint $c$ is violated. If there exist positive real numbers $\{\psi_v\}_{v \in V}$ such that for every variable $v \in V$,

$$
\frac{1}{\psi_v} \left( 1 + \sum_{c \ni v} P(A_c) \prod_{u \in c} \psi_u \right) < 1,
$$

then there exists a backtracking algorithm that finds a satisfying assignment efficiently with high probability.

We now use Theorem 6.1 to capture a well-known application of the Lovász Local Lemma to sparse $k$-SAT formulas when $P$ is the uniform measure. For a $k$-SAT formula $\Phi$ we will be denoting its maximum degree as $\Delta \equiv \Delta(\Phi)$, i.e., each variable of $\Phi$ is contained in at most $\Delta$ clauses.

**Theorem 6.2.** Every $k$-SAT formula $\Phi$ with maximum degree $\Delta < \frac{2^k}{ek}$ is satisfiable. Moreover, there exists a backtracking algorithm that finds a satisfying assignment of $\Phi$ efficiently.

**Proof.** Setting $\psi_v = \psi = 2\alpha > 0$ we see that it suffices to find a value $\alpha > 0$ such that

$$
\frac{1}{\psi} + \frac{1}{2^k} \Delta \psi^{k-1} = \frac{1}{2\alpha} + \frac{1}{2} \Delta \alpha^{k-1} < 1,
$$

which is feasible whenever

$$
\Delta < \max_{\alpha > 0} \frac{2\alpha - 1}{\alpha^k} = \frac{2^k}{k} \cdot \left( 1 - \frac{1}{k} \right)^{k-1} \leq \frac{2^k}{ek}.
$$

$\square$

**Remark 6.1.** In [32] it is shown that using a non-uniform product measure $P$ one can improve the bound of Theorem 6.2 to $\Delta < \frac{2^k}{ek+1}$ and that this is asymptotically tight. We note that we can achieve the same bound using Theorem 6.1 with the same $P$ but since this an involved LLL application we will not explicitly present it here.

6.1.1 Proof of Theorem 6.1

We consider a very simple backtracking algorithm: We start with each variable unassigned. Then, at each state $\sigma$ we choose the lowest indexed unassigned variable $v$ and sample a value for it according to the product measure $P$. If one or more constraints become violated, as a result we remove the value from each variable of the lowest indexed violated constraint.

Let $\Omega$ be the set of partial non-violating assignments. Let $\mu : \Omega \to \mathbb{R}$ be the probability measure that assigns to each state $\sigma \in \Omega$ the value $\mu(\sigma) \propto \prod_{v \in V} P(\sigma(v))$, where we abuse notation for brevity by letting $P(\sigma(v))$ denote the event that variable $v$ is assigned value $\sigma(v)$. We apply Theorem 3.3 using the diagonal matrix $M[\sigma, \sigma] = \mu(\sigma)$ and the norm $\| \cdot \|_1$. Crucially, we will be taking into account the observations of Section 3.3. Theorem 6.1 will follow immediately from the following lemma. (For brevity, we will index flaws with variables instead of integers.)
Lemma 6.3. For each vertex \( v \) and set of variables \( S \neq \emptyset \):

\[
\gamma_v^S = \begin{cases} 
1 & \text{if } S = \emptyset \\
P(A_c) & \text{if } S = c, \text{ where } c \text{ is a constraint containing } v \\
0 & \text{otherwise,}
\end{cases}
\]

Proof. Notice that the actions related to flaw \( f_v \) can only remove the value from sets of of variables that correspond to constraints that contain \( v \). Thus, \( \gamma_v^S = 0 \) for every set \( S \neq \emptyset \) that does not correspond to a constraint containing \( v \). Recalling the definition of the matrix \( C_v^S \) in Section 3.3 we have

\[
\gamma_v^S = \|MC_v^S M^{-1}\|_1 = \max_{\sigma' \in \Omega} \sum_{\sigma \in f_v} \frac{\mu(\sigma)}{\mu(\sigma')} \rho_v(\sigma, \sigma') .
\tag{26}
\]

To see the claim for the case of the empty set, notice that given a state \( \sigma' \) there exists at most one state \( \sigma \) such that \( \rho_v(\sigma, \sigma') > 0 \) and that \( U(\sigma') \setminus (U(\sigma) \setminus \{v\}) = \emptyset \). This is because we can uniquely reconstruct \( \sigma \) from \( \sigma' \) by removing the value from \( v \) at \( \sigma' \). Then we have

\[
\frac{\mu(\sigma)}{\mu(\sigma')} \rho_v(\sigma, \sigma') = \frac{\prod_{u \in V \setminus U(\sigma')} P(\sigma(u))}{\prod_{u \in V \setminus U(\sigma')} P(\sigma'(u))} P(\sigma'(v)) = \frac{1}{P(\sigma'(v))} P(\sigma'(v)) = 1 .
\]

To see the claim for the case where \( S = c \), consider the set \( \text{viol}(c) \) consisting of the set of value assignments of the variables of \( c \) that violate \( c \). Notice now that for every state \( \sigma' \in \Omega \) there is an injection from the set of states \( \sigma \) such that \( \rho_v(\sigma, \sigma') > 0 \) and \( S = U(\sigma') \setminus (U(\sigma) \setminus \{v\}) \) to \( \text{viol}(c) \). This is because \( c \) should be violated at each such state \( \sigma \) and, thus, it should be that each state \( \sigma \) should be of the form \( \sigma = \sigma'_\alpha \) for \( \alpha \in \text{viol}(c) \), where \( \sigma'_\alpha \) is the state induced by \( \sigma' \) when assigning \( \alpha \) to the variables of \( c \). Observe further that for every state of the form \( \sigma'_\alpha \), \( \alpha \in \text{viol}(c) \), we have that

\[
\frac{\mu(\sigma'_\alpha)}{\mu(\sigma')} \rho_v(\sigma'_\alpha, \sigma') = \left( \prod_{u \in c \setminus \{v\}} P(X_u = \sigma'_\alpha(u)) \right) P(X_v = \sigma'(v)) = P(A_c^0) ,
\tag{27}
\]

where \( P(A_c^0) \) is the probability of the event that the variables of \( c \) receive assignment \( \alpha \). Combining (27) with (26) and the fact that \( P(A_c) = \sum_{\alpha \in \text{viol}(c)} P(A_c^\alpha) \) concludes the proof of Lemma 6.3.

Plugging Lemma 6.3 into Theorem 3.3 concludes the proof of Theorem 6.1. As far as the running time is concerned, Corollary 3.7 implies that if

\[
T_0 = \log_2 \max_{\sigma \in \Omega} \frac{1}{\mu(\sigma)} + \log_2 \left( \prod_{u \in V} \psi_u \right) + \log_2 \left( \max_{S \subseteq [m]} \frac{1}{\prod_{v \in S} \psi_v} \right)
\]

then the probability that the algorithm makes \( \frac{T_0 + s}{\delta} \) steps is \( 2^{-s} \), where \( \delta = 1 - \max_{v \in V} \psi_v \).

6.2 Acyclic Edge Coloring

An edge-coloring of a graph is proper if all edges incident to each vertex have distinct colors. A proper edge coloring is acyclic if it has no bichromatic cycles, i.e., no cycle receives exactly two (alternating) colors. The smallest number of colors for which a graph \( G \) has an acyclic edge-coloring is denoted by \( \chi'_a(G) \).

Acyclic Edge Coloring was originally motivated by the work of Coleman et al. [23, 24] on the efficient computation of Hessians and, since then, there has been a series of papers [10, 54, 58, 38, 51, 30] that upper bound \( \chi'_a(G) \) for graphs with bounded degree. The current best result was given recently by Giotis et al. in [33] who showed that \( \chi'_a(G) \leq 3.74\Delta \) in graphs with maximum degree \( \Delta \).
6.2.1 A Simple Backtracking Algorithm

We show how one can apply Theorem 3.3 to recover the main application of the framework of [30] with a shorter proof.

Let $G$ be a graph with $m$ edges $E = \{e_1, \ldots, e_m\}$ and suppose we have $q$ available colors.

**Definition 6.4.** Given a graph $G = (V, E)$ and a (possibly partial) edge-coloring of $G$, say that color $e$ is 4-forbidden for $e \in E$ if assigning $e$ to $e$ would result in either a violation of proper-edge-coloration, or a bichromatic 4-cycle containing $e$. Say that $e$ is 4-available if it is not 4-forbidden.

**Lemma 6.5 ([30]).** In any proper edge-coloring of $G$ at most $2(\Delta - 1)$ colors are 4-forbidden for any $e \in E$.

**Proof.** The 4-forbidden colors for $e = \{u, v\}$ can be enumerated as: (i) the colors on edges adjacent to $u$, and (ii) for each edge $e_u$ adjacent to $v$, either the color of $e_v$ (if no edge with that color is adjacent to $u$), or the color of some edge $e'$ which together with $e, e_v$ and an edge adjacent to $u$ form a cycle of length 4. $\square$

Consider the following backtracking algorithm for Acyclic Edge Coloring with $q = 2(\Delta - 1) + Q$ colors. At each step, choose the lowest indexed uncolored edge $e$ and attempt to color it choosing uniformly at random among the 4-available colors for $e$. If one or more bichromatic cycles are created, then choose the lowest indexed one of them, say $C = \{e_1, e_2, \ldots, e_{2q} = e\}$, and remove the colors from all its edges except $e_1$ and $e_{2q}$.

The main result of [30] states the following.

**Theorem 6.6.** Every graph $G$ admits an acyclic edge coloring with $q > 4(\Delta - 1)$ colors. Moreover, such a coloring can be found in $O(|V| + |E|)$ time with high probability.

**Proof.** Let $\Omega$ be the set of partial acyclic edge colorings of $G$. For each edge $e$ let $f_e$ be the subset (flaw) of $\Omega$ that contains the partial acyclic edge colorings of $G$ in which $e$ is uncolored. We will apply Theorem 3.3 using the $\| \cdot \|_1$ norm and $M = \frac{1}{|\Omega|}$.

We first compute the charges $\gamma^S_e$ for each edge $e$ and set of edges $S$. Notice that for $\gamma^S_e$ to be non-zero, it should either be that $S = \emptyset$, or that $S$ contains $e$ and there exists a cycle $C = \{e_1, e_2\} \cup S$ so that, when a recoloring of $e$ makes $C$ bichromatic, the backtracking step uncolors precisely the edges in $S$. With that in mind, for each edge $e$ and each set $S$ that contains $e$, let $C_e(S)$ denote the set of cycles with the latter property.

**Lemma 6.7.** For each edge $e$, let

$$\gamma^S_e = \begin{cases} \frac{1}{Q} |C_e(S)| & \text{if } S = \emptyset \\ \frac{1}{Q} f_e(S) & \text{if } e \in S \\ 0 & \text{otherwise} \end{cases}$$

**Proof.** Notice that

$$\gamma^S_e = \max_{\sigma' \in \Omega} \sum_{\sigma \in f_e} \rho_e(\sigma, \sigma') \leq \max_{\sigma' \in \Omega} \sum_{\sigma \in f_e} \frac{1}{Q},$$

since according to Lemma 6.5, $\rho_e(\sigma, \sigma') \leq \frac{1}{Q}$ for each pair $(\sigma, \sigma') \in f_e \times \Omega$. The proof follows by observing that for each state $\sigma'$:

- If $S = \emptyset$ then there exists at most one state $\sigma$ such that $\rho_e(\sigma, \sigma') > 0$ and $U(\sigma') \setminus (U(\sigma) \setminus \{e\}) = \emptyset$ (we can reconstruct $\sigma$ from $\sigma'$ by uncoloring $e$).
• If \( S \ni e \) and \(|S| = 2\ell - 2 \) then there exist at most \(|C_e(S)|\) states such that \( \rho_e(\sigma, \sigma') > 0 \) and \( S = U(\sigma') \setminus (U(\sigma) \setminus \{e\}) \). Given a cycle \( C = S \cup \{e_{i_1}, e_{i_2}\} \) we reconstruct \( \sigma \) from \( \sigma' \) by finding the colors of edges in \( S \setminus \{e\} \) from \( \sigma'(e_{i_1}), \sigma'(e_{i_2}) \), exploiting the fact that the backtracking step corresponds to an uncoloring of a bichromatic cycle; \( e \) is uncolored; and every other edge has the same color as in \( \sigma' \).

• For all other \( S \) there exists no state \( \sigma \) such that \( \rho(\sigma, \sigma') > 0 \) and \( S = U(\sigma') \setminus (U(\sigma) \setminus \{e\}) \).

Observe that there are at most \((\Delta - 1)^{2\ell - 2}\) cycles of length \(2\ell\) containing a specific edge \( e \). In other words, there exist at most \((\Delta - 1)^{2\ell - 3}\) sets of edges \( S \) of size \(2\ell - 2\) that contain \( e \) and such that \( \gamma_e^S > 0 \) and, in addition, note that we always have \(|C_e(S)| \leq \Delta - 1\).

Thus, if \( Q = c(\Delta - 1) \) for some constant \( c \), setting \( \psi_e = \psi_e^c = \frac{\psi^3}{Q} \), where \( \psi \) is a constant in \((1, c)\), Lemma 6.7 implies:

\[
\frac{1}{\psi_e} \left( \sum_{S \subseteq E} \gamma_e^S \prod_{e \in S} \psi_j \right) \leq \min_{\psi \in (1, c)} \left( \frac{1}{\psi} + \sum_{i=3}^{\infty} \left( \frac{\Delta - 1}{Q} \right)^{2i-2} \psi^{2i-3} \right) \leq \min_{\psi \in (1, c)} \left( \frac{1}{\psi} + \frac{1}{c} \sum_{i=3}^{\infty} \left( \frac{\psi^3}{c} \right)^{2i-3} \right) = \min_{\psi \in (1, c)} \left( \frac{1}{\psi} + \frac{\psi^3}{c^2 (c^2 - \psi^2)} \right) = \frac{2}{c}
\]

for \( \psi^* = c \left( \frac{1 + \psi^2}{2} \right)^{-1} \). Thus, if \( c > 2 \) the probability that the algorithm fails to find an acyclic edge coloring within \( T_0 + s \) steps is \( 2^{-s} \), where \( \delta = 1 - \frac{2}{c} \), and, according to Corollary 3.7:

\[
T_0 = \log_2 |\Omega| + \log_2 (\psi^*)^{|E|} + \log_2 \left( \frac{1}{\psi^*} \right)^{|E|} = O(|V| + |E|) ,
\]

concluding the proof.

6.2.2 An Application of the Local Cut Lemma

Bernshteyn [13] introduced a non-constructive generalized LLL condition, called the “Local Cut Lemma”, with the aim of drawing connections between the LLL and the entropy compression method. He later applied it in [16] to the problem of Acyclic Edge Coloring giving improved bounds assuming further constraints on the graph besides sparsity. For example he proved the following.

**Theorem 6.8** ([16]). Let \( G \) be a graph with maximum degree \( \Delta \) and let \( H \) be a fixed bipartite graph. If \( G \) does not contain \( H \) as a subgraph, then there exists an acyclic edge coloring of \( G \) using at most \( 3(\Delta + o(1)) \) colors.

We now show how to use our framework to give a constructive proof of Theorem 6.8 i.e. we prove Theorem 5.8. This will follow immediately from the following structural lemma in [16].

**Lemma 6.9** ([16]). There exist positive constants \( \gamma, \delta \) such that the following holds. Let \( G \) be a graph with maximum degree \( \Delta \) that does not contain \( H \) as a subgraph. Then for any edge \( e \in E(G) \) and for any integer \( k \geq 4 \), the number of cycles of length \( k \) in \( G \) that contain \( e \) is at most \( \gamma \Delta^{k-2-\delta} \).
\textbf{Proof of Theorem 3.8} Notice that in this case, making almost identical calculations to those above, invoking Lemma 6.9 to upper bound the number of cycles that contain \(e\) and setting \(\psi = \frac{1}{\alpha}\) we obtain
\[
\frac{1}{\psi_e} \left( \sum_{S \subseteq E} \gamma_e^S \prod_{h \in S} \psi_h \right) \leq \min_{\psi \in (1, c)} \left( \frac{1}{\psi} + (\psi)^3 \gamma \Delta^{-\delta} \right) = \frac{1}{c} \min_{\alpha > 1} \left( \alpha + \frac{\alpha \gamma \Delta^{-\delta}}{\alpha (\alpha^2 - 1)} \right).
\]

Thus, as \(\Delta\) grows, the value of \(c\) required for the algorithm to terminate approaches 1, concluding the proof. \(\square\)

\section{Proof of the Witness Tree Lemma for Commutative Algorithms}

In this section we state and prove the Witness Tree Lemma for our setting. To do so, we will first need to recall the definition of witness trees from [57] (slightly reformulated to fit our setting).

For a trajectory \(\Sigma = \sigma_1 \ldots \sigma_t\), where \(\sigma_1 \ldots \sigma_t\) denotes that flaw \(f_{w_i}\) was addressed at the \(i\)-th step, we define the witness, \(W(\Sigma)\), of \(\Sigma\) to be \((w_1, \ldots, w_t)\). A witness tree \(\tau = (T, \ell_T)\) is a finite rooted, unordered tree \(T\) along with a labelling \(\ell_T : V(T) \rightarrow \{1, 2, \ldots, m\}\) of its vertices with indices of flaws such that the children of a vertex \(v \in V(T)\) receive labels from \(N(\ell(v))\). To lighten the notation, we sometimes write \([v]\) to denote \(\ell(v)\) and \(V(\tau)\) instead of \(V(T)\). Given a witness \(W = (w_1, w_2, \ldots, w_t)\) we associate with each \(i \in \{1, 2, \ldots, t\}\) a witness tree \(\tau_W(i)\) that is constructed as follows: Let \(\tau_W^{(i)}(i)\) be an isolated vertex labelled by \(w_i\). Then, going backwards for each \(j = i - 1, i - 2, \ldots, 1\): if there is a vertex \(v \in \tau_W^{(j+1)}(i)\) such that \([v] \sim w_j\) then we choose among those vertices the one having the maximum distance (breaking ties arbitrarily) from the root and attach a new child vertex \(u\) to \(v\) that we label \(w_j\) to get \(\tau_W^{(j)}(i)\).

If there is no such vertex \(v\) then \(\tau_W^{(j+1)}(i) = \tau_W^{(j)}(i)\). Finally, let \(\tau_W(i) = \tau_W^{(1)}(i)\).

We will say that a witness tree \(\tau\) occurs in a trajectory \(\Sigma\) if \(W(\Sigma) = (w_1, w_2, \ldots, w_t)\) and there is \(k \in \{1, 2, \ldots, t\}\) such that \(\tau_W(k) = \tau\). For an arbitrary permutation \(\pi\) of \(\{1, 2, \ldots, m\}\) let \(\chi_\pi(\tau)\) be the ordered witness tree that is induced by ordering the children of each node in \(\tau\) from left to right, increasingly according to \(\pi\).

Finally, recall that \(\theta\) denotes the initial distribution of our algorithm and for \(i \in \{1, 2, \ldots, m\}\) define
\[
\gamma_i = \|MA_iM^{-1}\|, \quad M \text{ is a fixed invertible matrix.}
\]

\textbf{Theorem 7.1} (Witness Tree Lemma). \textit{Assume that algorithm} \(A\) \textit{is commutative with respect to binary relation~ and follows a canonical flaw choice strategy. Let} \(\pi\) \textit{be an arbitrary permutation of} \(\{1, 2, \ldots, m\}\). \textit{Then, for every witness tree} \(\tau\),
\[
\Pr[\tau] \leq \lambda_{\text{init}} \prod_{i=1}^{|\tau|} \gamma_{\chi_\pi(i)},
\]
\textit{where} \((v_1, v_2, \ldots, v_{|\tau|})\) \textit{are the vertices of} \(\chi_\pi(\tau)\) \textit{in backward breadth first order and} \(\lambda_{\text{init}} = \|\theta M^{-1}\|\).

Finally, we prove a structural property of the witness trees that might occur during the execution of the algorithm which will be helpful in our analysis.

\textbf{Proposition 7.1.} \textit{For a witness tree} \(\tau = (T, \ell_T)\) \textit{let} \(L_i = L_i(\tau)\) \textit{denote the set of labels of the nodes at distance} \(i\) \textit{from the root. For each} \(i \geq 0\), \textit{and each} \(\alpha, \beta \in L_i\) \textit{we have that} \(\alpha \sim \beta\).
Proof. Let \( W = (w_1, w_2, \ldots, w_t) \) be a witness sequence that can occur in an execution of our algorithm. Let \( \alpha, \beta \) be two distinct elements of \( L_i \). By the definition of \( \tau \), labels \( \alpha, \beta \) correspond to two indices \( w_{j_1}, w_{j_2} \) of \( W \). Assume without loss of generality that \( j_1 < j_2 \). Then, according to the algorithm for constructing \( \tau \), index \( w_{j_2} \) is “attached first” to the \( i \)-th level of \( \tau \). The proof is concluded by noticing that if \( w_{j_1} = \alpha \sim \beta = w_{j_2} \) then the node corresponding to \( w_{j_2} \) is eligible to be a child of the node corresponding to \( w_{j_1} \) during the construction of \( \tau \) and, thus, \( \beta \notin L_i \), which is a contradiction. \( \square \)

7.1 Proof of Theorem 7.1

We will assume without loss of generality that our algorithm follows a deterministic canonical flaw choice strategy. This is because randomized flaw choice strategies can equivalently be interpreted as convex combination of deterministic ones, and hence their analysis is equivalent to taking an expectation over deterministic strategies.

Theorem 7.1 will follow from Lemmas 7.2 and 7.3.

Lemma 7.2. Let \( A \) be a commutative algorithm with respect to \( \sim \). Then for each witness \( W = (w_1, \ldots, w_t) \),

\[
\Pr \left[ \text{The witness of the trajectory of } A \text{ has } W \text{ as prefix} \right] \leq \left\| \theta^T \prod_{i=1}^t A_{w_i} \right\|_1 .
\]

Lemma 7.3. Let \( A = A(S) \) be a commutative algorithm with respect to \( \sim \) that follows a canonical flaw choice strategy \( S \) and let \( \tau \) be a witness tree. Then there exists a canonical flaw choice strategy \( S' \) such that, for algorithm \( A' = A(S') \),

\[
\Pr_{A}[\tau] \leq \Pr_{A'}[\text{The witness of the trajectory of } A' \text{ has } ([v_1], [v_2], \ldots, [v_{|\tau|}]) \text{ as prefix}] ,
\]

where \( v_1, v_2, \ldots, v_{|\tau|} \) are the vertices of \( \chi_{\pi}(\tau) \) in backward breadth first order.

Proof of Theorem 7.1 Define \( q = \theta^T \prod_{i=1}^{|\tau|} A_{[v_i]} \) and let \( A' \) be the algorithm promised by Lemma 7.3 when applied to \( A \). Applying Lemmata 7.2 and 7.3 we obtain

\[
\Pr[\tau] \leq \Pr \left[ \text{The witness of the trajectory of } A' \text{ has } ([v_1], \ldots, [v_{|\tau|}]) \text{ as prefix} \right] \leq \sum_{\sigma' \in \Omega} q(\sigma') . \tag{28}
\]

Now the righthand side of (28) can be written as

\[
\sum_{\sigma' \in \Omega} \theta^T M^{-1} \left( \prod_{i=1}^{|\sigma|} MA_{[v_i]} M^{-1} \right) M e_{\sigma'} \leq \left\| \theta^T M^{-1} \prod_{i=1}^{|\sigma|} MA_{[v_i]} M^{-1} \right\| \sum_{\sigma' \in \Omega} \left\| Me_{\sigma'} \right\| \tag{29}
\]

\[
\leq \| \theta^T M^{-1} \| \cdot \prod_{i=1}^{|\sigma|} \left\| MA_{[v_i]} M^{-1} \right\| \tag{30}
\]

\[
= \lambda_{\text{init}} \prod_{i=1}^{|\tau|} \gamma_{[v_i]} , \tag{31}
\]

where to get (29) we apply (28), to get (30) we first apply (28) and then (27) and that \( \sum_{\sigma' \in \Omega} \| Me_{\sigma'} \| = 1 \), and, finally, (31) holds by the definitions of \( \lambda_{\text{init}} \) and \( \gamma_i \). Note that (31) concludes the proof. \( \square \)
7.1.1 Proof of Lemma 7.2

Given a deterministic flaw choice strategy \( S = (s_1, s_2, \ldots) \), an integer \( t \geq 1 \) and \( i \in \{1, 2, \ldots, m\} \), define matrix \( A_{i,s_t} \) by \( A_{i,s_t}[^{[\sigma, \sigma']}] = A_i[^{[\sigma, \sigma']} \rho_i[^{[\sigma, \sigma']} \) if \( s_t[^{[\sigma]} = i \) and \( A_{i,s_t}[^{[\sigma, \sigma']} = 0 \) otherwise. Moreover, let \( P(W) \) denote the set of witnesses that can occur in an execution of \( A \) and have \( W = (w_1, w_2, \ldots, w_t) \) as prefix.

With this notation, we can express the probability that the witness of the trajectory of \( A \) has \( W \) as a prefix as

\[
\left\| \theta^T \prod_{i=1}^{t} A_{w_i,s_i} \sum_{W' \in P(W)} \prod_{j=t+1}^{W'} A_{w'_j,s'_j} \right\|_1 .
\]

(32)

Define \( q = \theta^T \prod_{i=1}^{t} A_{w_i,s_i} \), and for any state \( \sigma \) let \( P_\sigma(W) \) denote the subset of witnesses of \( P(W) \) that have \( s_{t+1}(\sigma) \) as their \( (t+1) \) element. With this notation we can write (32) as

\[
\sum_{\sigma \in \Omega} \sum_{\sigma' \in \Omega} q(\sigma') e_{\sigma'} \sum_{W' \in P(W)} \prod_{j=t+1}^{W'} A_{w'_j,s'_j} e_{\sigma} = \sum_{\sigma \in \Omega} \sum_{\sigma' \in \Omega} q(\sigma') \sum_{W' \in P_\sigma'(W)} \prod_{j=t+1}^{W'} A_{w'_j,s'_j} e_{\sigma} \\
\leq \sum_{\sigma' \in \Omega} q(\sigma') \sum_{\sigma \in \Omega} e_{\sigma'} \sum_{W' \in P_\sigma(W)} \prod_{j=t+1}^{W'} A_{w'_j,s'_j} e_{\sigma} \\
\leq \sum_{\sigma' \in \Omega} q(\sigma') \leq \left\| \theta^T \prod_{i=1}^{t} A_{w_i} \right\|_1 ,
\]

where the first equality follows from the fact that at step \( t+1 \) the algorithm addresses flaw \( f_{s_{t+1}(\sigma')} \), and the last inequality follows from the fact that

\[
\sum_{\sigma \in \Omega} \sum_{W' \in P_\sigma(W)} \prod_{j=t+1}^{W'} A_{w'_j,s'_j} e_{\sigma} = \left\| \prod_{j=t+1}^{W'} A_{w'_j,s'_j} e_{\sigma} \right\|_1 \\
\]

is at most the probability that \( A \) starts from state \( \sigma' \) and ends at some state in \( \Omega \) and is therefore at most 1.

7.1.2 Proof of Lemma 7.3

Recall the definition of canonical flaw choice strategies \( S = (s_1, s_2, \ldots) \), as well as the definition of \( A_{i,s_t} \) for \( i \in \{1, 2, \ldots, m\} \) and \( t \geq 1 \). Let \( W(\tau) \) denote the set of witnesses that can occur in a trajectory of \( A(S) \) and for which \( \tau \) occurs. That is, for each \( W \in W(\tau) \) there exists \( t \in \{1, \ldots, |W|\} \) so that \( \tau_W(t) = \tau \). We can express the probability of \( \tau \) occurring in an execution of \( A \) as

\[
\Pr[\tau] = \left\| \sum_{W \in W(\tau)} \theta^T \prod_{i=1}^{W} A_{w_i,s_i} \right\|_1 .
\]

The idea now will be to gradually transform \( S \) and \( W(\tau) \) to the canonical flaw choice strategy \( S' = (s'_1, s'_2, \ldots) \) and set \( W'(\tau) \), respectively, so that every witness in \( W'(\tau) \) has \( (v[1], v[2], \ldots, v[\tau]) \) as a prefix, and

\[
\Pr[\tau] = \left\| \sum_{W \in W(\tau)} \theta^T \prod_{i=1}^{W} A_{w_i,s_i} \right\|_1 \leq \left\| \sum_{W' \in W'(\tau)} \theta^T \prod_{i=1}^{W'} A_{w'_i,s'_i} \right\|_1 .
\]

(33)
This will suffice to prove the lemma since \( \mathcal{A}' = \mathcal{A}(S') \) we have that
\[
\left\| \sum_{W' \in \mathcal{W}'(\tau)} \theta T \prod_{i=1}^{[\mathcal{W}']} A_{w_i,s_i} \right\|_1 \leq P_{\mathcal{A}'} \left( \text{The witness of the trajectory of } \mathcal{A}' \text{ has } ([v_1], [v_2], \ldots, [v_{|\tau|}] \text{ as prefix} \right).
\]

We first define the elementary operation for transforming \((\mathcal{S}, \mathcal{W}(\tau))\) to \((\mathcal{S}', \mathcal{W}'(\tau))\). For \( S \subseteq \{1, \ldots, m\} \) recall that we define \( \pi(S) \) to be the lowest indexed integer according to \( \pi \) in \( S \). Given an integer \( 1 \leq i \leq m \) we define \( p_i : \Omega \to \{1, 2, \ldots, m\} \) to be the following function. For every state \( \sigma \):
\[
p_i(\sigma) = \begin{cases} 
  i & \text{if } \sigma \in f_i, \\
  \pi(U(\sigma)) & \text{otherwise}.
\end{cases}
\]

In words, \( p_i(\sigma) \) always gives priority to flaw \( f_i \), unless \( f_i \) is not present in \( \sigma \) in which case it selects the flaw corresponding to the lowest index in \( U(\sigma) \). We also define function \( \text{Swap} \) that takes as input a canonical flaw choice strategy \( S_1 = (s_1, \ldots, s_{i_1}, s_{i_2}, \ldots) \), a set of witnesses \( \mathcal{W}_1 \), a witness \( W = (w_1, w_2, \ldots w_t) \in \mathcal{W}_1 \) and an integer \( i \in \{1, 2, \ldots, |\mathcal{W}|\} \) such that \( w_{i-1} \sim w_i \), and outputs \((S_2, \mathcal{W}_2)\) defined as follows:

- \( S_2 = (s_1, \ldots, p_{w_1}, p_{w_{i-1}}, s_{i+1}, \ldots) \);
- \( \mathcal{W}_2 \) is obtained from \( \mathcal{W}_1 \) by changing every witness \( W' = (w'_1, w'_2, \ldots w'_{i-1}, w'_i, \ldots) \in \mathcal{W}_1 \) such that \( w'_i = w_i \) and \( w'_{i-1} = w_{i-1} \) to \( W'' = (w'_1, w'_2, \ldots w'_i, \ldots) \).

We now describe the algorithm that achieves the transformation of \((\mathcal{S}, \mathcal{W})\) to \((\mathcal{S}', \mathcal{W}')\) and then prove its correctness. For a witness sequence \( W \in \mathcal{W}(\tau) \) and a vertex \( v_i \) of \( \tau \), let \( p_W(v_i) \) denote the position of the element that corresponds to \([v_i]\) in \( W \). The algorithm is as follows.

1. Set \( \mathcal{S}' \leftarrow S \) and \( \mathcal{W}'(\tau) \leftarrow \mathcal{W}(\tau) \)
2. For \( i = 1 \) to \(|\tau|\)
   - While there exist \( W \in \mathcal{W}'(\tau) \) for which \( p_W(v_i) \neq i \)
     - \((S', \mathcal{W}'(\tau)) \leftarrow \text{Swap}(S', \mathcal{W}'(\tau), W, p_W(v_i))\)

First notice that if the algorithm terminates then each witness \( W \in \mathcal{W}'(\tau) \) has \(([v_1], [v_2], \ldots, [v_{|\tau|}])\) as a prefix. Moreover, it always terminates because at every step, if \( W = (w_1, \ldots, w_2) \) is the input of \( \text{Swap} \), then \( w_{p_W(v_i)-1} \sim w_{p_W(v_i)} \). To see this, observe that the fact that there exists \( t \in \{1, 2, \ldots, |\mathcal{W}|\} \) such that \( \tau_W(t) = \pi_W(t) = \pi(t) \) implies that \( f_{w_{p_W(v_i)-1}} \) is either a flaw that appears in \( \tau \) and corresponds to a vertex at the same level with \( f_{w_{p_W(v_i)}} \) or doesn’t appear in \( \tau \). Proposition 7.1 and the definition of the algorithm for constructing witness trees guarantee that in both these cases we have \( w_{p_W(v_i)-1} \sim w_{p_W(v_i)} \).

Finally, we need to show that \((\mathcal{S}', \mathcal{W}', \mathcal{W}^*) = \text{Swap}(\mathcal{S}, \mathcal{W}, \mathcal{W}^*, k)\) Then,
\[
\left\| \sum_{W \in \mathcal{W}(\tau)} \theta^T \prod_{i=1}^{[\mathcal{W}']} A_{w_i,s_i} \right\|_1 \leq \left\| \sum_{W' \in \mathcal{W}'(\tau)} \theta^T \prod_{i=1}^{[\mathcal{W}']} A_{w'_i,s'_i} \right\|_1.
\]

Lemma 7.4. Fix any set of witnesses \( \mathcal{W} \) and canonical flaw choice strategy \( \mathcal{S} \). For some \( W^* \in \mathcal{W} \) and \( k \in \{1, \ldots, |\mathcal{W}^*|\} \), let \((\mathcal{S}', \mathcal{W}') = \text{Swap}(\mathcal{S}, \mathcal{W}, \mathcal{W}^*, k)\). Then,
We start with a key observation.

Lemma 7.5. Consider an arbitrary pair of functions \( s_1, s_2 : \Omega \to \{1, \ldots, m\} \) that take as input a state \( \sigma \) and output an index in \( U(\sigma) \). Let \( i \sim j \) be two arbitrary indices in \( \{1, \ldots, m\} \). Then, for every pair of states \( \sigma_1, \sigma_2 \in \Omega \) we have that \( A_{i,s_1} A_{j,s_2}[\sigma_1, \sigma_2] \leq A_{j,p_j} A_{i,p_i}[\sigma_1, \sigma_2] \).

Proof:

\[
A_{i,s_1} A_{j,s_2}[\sigma_1, \sigma_2] \leq \sum_{\sigma \in A(i, \sigma_1)} \rho_i(\sigma_1, \sigma) \rho_j(\sigma, \sigma_2) = \sum_{\sigma \in A(j, \sigma_1)} \rho_j(\sigma_1, \sigma) \rho_i(\sigma, \sigma_2) = A_{j,p_j} A_{i,p_i}[\sigma_1, \sigma_2]
\]

where the first equality holds because \( A_i A_j = A_j A_i \) (since \( i \sim j \)) and the second equality holds by the definitions of \( p_i, p_j \).

Let \( W^* = (w^*_1, w^*_2, \ldots, w^*_{i-1}, w^*_i, \ldots) \) and define \( A(W^*) \subseteq W \) to be the set of witnesses that will be affected by \( \text{Swap} \), i.e., the set of witnesses whose \((i-1)\)-th and \(i\)-th elements are \( w^*_{i-1} \) and \( w^*_i \), respectively. Finally, let \( I_A(W^*) \) be the subset of \( \mathcal{W}' \) that is the image of \( A(W^*) \). We consider two cases.

In the first case, we assume that the mapping from \( \mathcal{W} \) to \( \mathcal{W}' \) is bijective. Given a witness sequence \( W = (w_1, \ldots, w^*_{i-1}, w^*_i, \ldots, w_t) \in A(W^*) \), let \( W' = (w_1, \ldots, w^*_{i-1}, w^*_i, \ldots, w_t) \in I_A(W^*) \) be the corresponding witness according to the bijection. By Lemma 7.5 and the definition of \( S' \) we obtain

\[
\| \theta^T A_{w_1,s_1} \cdots A_{w_{i-1},s_{i-1}} A_{w_i,s_i} \cdots \|_1 \leq \| \theta^T A_{w_1,p_{w_1}} A_{w_{i-1},p_{w_{i-1}}} \cdots \|_1,
\]

which concludes the proof for this case.

In the second case, the mapping from \( \mathcal{W} \) to \( \mathcal{W}' \) is not bijective. Observe that this can only be the case when there exist pairs \( (W_1, W_2) \in \mathcal{W} \) of the form

\[
W_1 = (w_1, w_2, \ldots, w^*_{i-1}, w^*_i, \ldots, w_t);
W_2 = (w_1, w_2, \ldots, w^*_i, w^*_i, \ldots, w_t),
\]

in which case \( W_1 \) is mapped to \( W_2 \). Thus, it suffices to show that

\[
\| \theta^T A_{w_1,s_1} \cdots (A_{w^*_{i-1},s_{i-1}} A_{w^*_i,s_i} + A_{w^*_i,s_i} A_{w^*_{i-1},s_i}) \cdots A_{w_t,s_t} \|_1 \leq \| \theta^T A_{w_1,p_{w_1}} A_{w^*_{i-1},p_{w_{i-1}}} A_{w^*_i,p_{w_i}} \cdots A_{w_t,s_t} \|_1.
\]

(34)

To prove (34) it suffices to show that

\[
A_{w^*_{i-1},s_{i-1}} A_{w^*_i,s_i}[\sigma, \sigma'] + A_{w^*_i,s_i} A_{w^*_{i-1},s_i}[\sigma, \sigma'] \leq A_{w^*_i,p_{w_i}} A_{w^*_{i-1},p_{w_{i-1}}}[\sigma, \sigma'],
\]

(35)

for each pair of states \((\sigma, \sigma')\). If \( s_{i-1}(\sigma) \notin \{w^*_i, w^*_{i-1}\} \), then (35) is trivially true. If, on the other hand, \( s_{i-1}(\sigma) \in \{w^*_i, w^*_{i-1}\} \), then either \( s_{i-1}(\sigma) = w^*_i \) and \( A_{w^*_{i-1},s_{i-1}} A_{w^*_i,s_i}[\sigma, \sigma'] = 0 \), or \( s_{i-1}(\sigma) = w^*_{i-1} \) and \( A_{w^*_{i-1},s_{i-1}} A_{w^*_i,s_i}[\sigma, \sigma'] = 0 \). The proof of (35) in the first case follows immediately by the definitions of \( p_{w_{i-1}}, p_{w_i} \) and in the second case from Lemma 7.5.

8 Acknowledgements

We are grateful to Paris Syminelakis for various insightful remarks and comments. We also thank David Harris for helpful discussions.
References

[1] Dimitris Achlioptas and Amin Coja-Oghlan. Algorithmic barriers from phase transitions. In 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA, pages 793–802. IEEE Computer Society, 2008.

[2] Dimitris Achlioptas, Themis Gouleakis, and Fotis Iliopoulos. Local computation algorithms for the Lovász Local Lemma. CoRR, abs/1809.07910, 2018.

[3] Dimitris Achlioptas and Fotis Iliopoulos. Random walks that find perfect objects and the Lovász local lemma. In 55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014, pages 494–503. IEEE Computer Society, 2014.

[4] Dimitris Achlioptas and Fotis Iliopoulos. Random walks that find perfect objects and the Lovász local lemma. J. ACM, 63(3):22:1–22:29, July 2016.

[5] Dimitris Achlioptas, Fotis Iliopoulos, and Vladimir Kolmogorov. A local lemma for focused stochastic algorithms. arXiv preprint arXiv:1809.01537, 2018.

[6] Dimitris Achlioptas, Fotis Iliopoulos, and Nikos Vlassis. Stochastic control via entropy compression. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland, volume 80 of LIPIcs, pages 83:1–83:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[7] Dimitris Achlioptas and Cristopher Moore. Random k-sat: two moments suffice to cross a sharp threshold. SIAM Journal on Computing, 36(3):740–762, 2006.

[8] Dimitris Achlioptas and Assaf Naor. The two possible values of the chromatic number of a random graph. In László Babai, editor, Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004, pages 587–593. ACM, 2004.

[9] Michael Albert, Alan Frieze, and Bruce Reed. Multicoloured hamilton cycles. the electronic journal of combinatorics, 2(1):R10, 1995.

[10] Noga Alon. A parallel algorithmic version of the local lemma. Random Structures & Algorithms, 2(4):367–378, 1991.

[11] Noga Alon, Michael Krivelevich, and Benny Sudakov. Coloring graphs with sparse neighborhoods. Journal of Combinatorial Theory, Series B, 77(1):73–82, 1999.

[12] Noga Alon and Joel H. Spencer. The Probabilistic Method. Wiley Publishing, 4th edition, 2016.

[13] Rogério G Alves, Aldo Procacci, and Remy Sanchis. A local lemma via entropy compression. arXiv preprint arXiv:1509.04638, 2015.

[14] Adrian Balint and Uwe Schöning. Choosing probability distributions for stochastic local search and the role of make versus break. In International Conference on Theory and Applications of Satisfiability Testing, pages 16–29. Springer, 2012.

[15] Adrian Balint and Uwe Schöning. Engineering a lightweight and efficient local search SAT solver. In Lasse Kliemann and Peter Sanders, editors, Algorithm Engineering - Selected Results and Surveys, volume 9220 of Lecture Notes in Computer Science, pages 1–18. 2016.
[16] Anton Bernshteyn. New bounds for the acyclic chromatic index. *Discrete Mathematics*, 339(10):2543–2552, 2016.

[17] Anton Bernshteyn. The johansson–molloy theorem for dp-coloring. *arXiv preprint arXiv:1708.03843*, 2017.

[18] Anton Bernshteyn. The local cut lemma. *European Journal of Combinatorics*, 63:95–114, 2017.

[19] Rodrigo Bissacot and Luís Doin. Entropy compression method and legitimate colorings in projective planes. *arXiv preprint arXiv:1710.06981*, 2017.

[20] Rodrigo Bissacot, Roberto Fernández, Aldo Procacci, and Benedetto Scoppola. An improvement of the Lovász local lemma via cluster expansion. *Combinatorics, Probability & Computing*, 20(5):709–719, 2011.

[21] Bartłomiej Bosek, Sebastian Czerwiński, Jarosław Grytczuk, and Paweł Rzążewski. Harmonious coloring of uniform hypergraphs. *Applicable Analysis and Discrete Mathematics*, 10(1):73–87, 2016.

[22] Karthkeyan Chandrasekaran, Navin Goyal, and Bernhard Haeupler. Deterministic algorithms for the Lovász local lemma. *SIAM J. Comput.*, 42(6):2132–2155, 2013.

[23] Kai-Min Chung, Seth Pettie, and Hsin-Hao Su. Distributed algorithms for the Lovász local lemma and graph coloring. In Magnús M. Halldórsson and Shlomi Dolev, editors, *ACM Symposium on Principles of Distributed Computing, PODC ’14, Paris, France, July 15-18, 2014*, pages 134–143. ACM, 2014.

[24] Thomas F Coleman and Jin Yi Cai. The cyclic coloring problem and estimation of spare hessian matrices. *SIAM J. Algebraic Discrete Methods*, 7(2):221–235, April 1986.

[25] Thomas F. Coleman and Moré Jorge J. Estimation of sparse hessian matrices and graph coloring problems. *Mathematical Programming*, 28(3):243–270, 1984.

[26] Andrzej Dudek, Alan Frieze, and Andrzej Ruciński. Rainbow hamilton cycles in uniform hypergraphs. *the electronic journal of combinatorics*, 19(1):46, 2012.

[27] Vida Dujmović, Gwenaël Joret, Jakub Kozik, and David R Wood. Nonrepetitive colouring via entropy compression. *Combinatorica*, 36(6):661–686, 2016.

[28] Paul Erdős and László Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In *Infinite and finite sets (Colloq., Keszthely, 1973; dedicated to P. Erdős on his 60th birthday), Vol. II*, pages 609–627. Colloq. Math. Soc. János Bolyai, Vol. 10. North-Holland, Amsterdam, 1975.

[29] Paul Erdős and Joel Spencer. Lopsided Lovász local lemma and latin transversals. *Discrete Applied Mathematics*, 30(2-3):151–154, 1991.

[30] Louis Esperet and Aline Parreau. Acyclic edge-coloring using entropy compression. *European Journal of Combinatorics*, 34(6):1019–1027, 2013.

[31] Iliopoulos Fotis and Sinclair Alistair. Efficiently list-edge coloring multigraphs asymptotically optimally. *Manuscript*, 2018.

[32] Heidi Gebauer, Tibor Szabó, and Gábor Tardos. The local lemma is tight for SAT. In Dana Randall, editor, *SODA*, pages 664–674. SIAM, 2011.
[33] Ioannis Giotis, Lefteris M. Kirousis, Kostas I. Psaromiligkos, and Dimitrios M. Thilikos. Acyclic edge coloring through the Lovász local lemma. *Theor. Comput. Sci.*, 665:40–50, 2017.

[34] Adam Gkagol, Gwenael Joret, Jakub Kozik, and Piotr Micek. Pathwidth and nonrepetitive list coloring. *arXiv preprint arXiv:1601.01886*, 2016.

[35] Daniel Gonçalves, Mickaël Montassier, and Alexandre Pinlou. Entropy compression method applied to graph colorings. *arXiv preprint arXiv:1406.4380*, 2014.

[36] Jarosław Grytczuk, Jakub Kozik, and Piotr Micek. New approach to nonrepetitive sequences. *Random Structures & Algorithms*, 42(2):214–225, 2013.

[37] Bernhard Haeupler and David G Harris. Parallel algorithms and concentration bounds for the Lovász local lemma via witness-DAGs. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1170–1187. SIAM, 2017.

[38] Bernhard Haeupler, Barna Saha, and Aravind Srinivasan. New constructive aspects of the Lovász local lemma. *J. ACM*, 58(6):Art. 28, 2011.

[39] David G. Harris. Lopsidependency in the Moser-Tardos framework: Beyond the lopsided Lovász local lemma. *ACM Trans. Algorithms*, 13(1):17:1–17:26, 2016.

[40] David G. Harris. Comparison for two convergence criteria for the variable-assignment lopsided lovasz local lemma. *arXiv preprint arXiv:1610.01926*, 2018.

[41] David G. Harris and Aravind Srinivasan. The Moser-Tardos framework with partial resampling. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013*, Berkeley, CA, USA, pages 469–478. IEEE Computer Society, 2013.

[42] David G. Harris and Aravind Srinivasan. A constructive algorithm for the Lovász local lemma on permutations. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*, Portland, Oregon, USA, January 5-7, 2014, pages 907–925. SIAM, 2014.

[43] David G. Harris and Aravind Srinivasan. Algorithmic and enumerative aspects of the Moser-Tardos distribution. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016*, Arlington, VA, USA, January 10-12, 2016, pages 2004–2023, 2016.

[44] Nicholas J. A. Harvey and Jan Vondrák. An algorithmic proof of the Lovász local lemma via resampling oracles. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015*, Berkeley, CA, USA, 17-20 October, 2015, pages 1327–1346. IEEE Computer Society, 2015.

[45] Kun He, Liang Li, Xingwu Liu, Yuqi Wang, and Mingji Xia. Variable-version lovász local lemma: Beyond shearer’s bound. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pages 451–462. IEEE, 2017.

[46] Fotis Iliopoulos. Commutative algorithms approximate the LLL-distribution. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018*, August 20-22, 2018 - Princeton, NJ, USA, volume 116 of *LIPIcs*, pages 44:1–44:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.
[47] A. Johansson. Asympotic choice number for triangle free graphs. Unpublished manuscript, 1996.

[48] Jeff Kahn. Asymptotics of the chromatic index for multigraphs. Journal of Combinatorial Theory, Series B, 68(2):233–254, 1996.

[49] Jeff Kahn. Asymptotics of the list-chromatic index for multigraphs. Random Structures & Algorithms, 17(2):117–156, 2000.

[50] Kashyap Babu Rao Kolipaka and Mario Szegedy. Moser and Tardos meet Lovász. In STOC, pages 235–244. ACM, 2011.

[51] Kashyap Babu Rao Kolipaka, Mario Szegedy, and Yixin Xu. A sharper local lemma with improved applications. In Anupam Gupta, Klaus Jansen, José D. P. Rolim, and Rocco A. Servedio, editors, Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings, volume 7408 of Lecture Notes in Computer Science, pages 603–614. Springer, 2012.

[52] Vladimir Kolmogorov. Commutativity in the algorithmic Lovász local lemma. In Irit Dinur, editor, IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA, pages 780–787. IEEE Computer Society, 2016.

[53] Michael Molloy. The list chromatic number of graphs with small clique number. Journal of Combinatorial Theory, Series B, 2018.

[54] Michael Molloy and Bruce Reed. Further algorithmic aspects of the local lemma. In STOC ’98 (Dallas, TX), pages 524–529. ACM, New York, 1999.

[55] Michael Molloy and Bruce Reed. Graph colouring and the probabilistic method, volume 23 of Algorithms and Combinatorics. Springer-Verlag, Berlin, 2002.

[56] Robin A. Moser. A constructive proof of the Lovász local lemma. In STOC’09—Proceedings of the 2009 ACM International Symposium on Theory of Computing, pages 343–350. ACM, New York, 2009.

[57] Robin A. Moser and Gábor Tardos. A constructive proof of the general Lovász local lemma. J. ACM, 57(2):Art. 11, 15, 2010.

[58] Sokol Ndreca, Aldo Procacci, and Benedetto Scoppola. Improved bounds on coloring of graphs. Eur. J. Comb., 33(4):592–609, May 2012.

[59] Pascal Ochem and Alexandre Pinlou. Application of entropy compression in pattern avoidance. arXiv preprint arXiv:1301.1873, 2013.

[60] Wesley Pegden. An extension of the Moser-Tardos algorithmic local lemma. SIAM J. Discrete Math., 28(2):911–917, 2014.

[61] Jakub Przybyło. On the facial thue choice index via entropy compression. Journal of Graph Theory, 77(3):180–189, 2014.

[62] Alexander D Scott and Alan D Sokal. The repulsive lattice gas, the independent-set polynomial, and the lovász local lemma. Journal of Statistical Physics, 118(5-6):1151–1261, 2005.
[63] Bart Selman, Henry A. Kautz, and Bram Cohen. Local search strategies for satisfiability testing. In David S. Johnson and Michael A. Trick, editors, Cliques, Coloring, and Satisfiability, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, October 11-13, 1993, volume 26 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pages 521–532. DIMACS/AMS, 1993.

[64] Bart Selman, Henry A. Kautz, and Bram Cohen. Noise strategies for improving local search. In Barbara Hayes-Roth and Richard E. Korf, editors, Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, WA, USA, July 31 - August 4, 1994, Volume 1., pages 337–343. AAAI Press / The MIT Press, 1994.

[65] J.B. Shearer. On a problem of Spencer. Combinatorica, 5(3):241–245, 1985.

[66] Van H Vu. A general upper bound on the list chromatic number of locally sparse graphs. Combinatorics, Probability and Computing, 11(1):103–111, 2002.
A Matrices and Norms

Let \( \| \cdot \| \) be any norm over vectors in \( \mathbb{R}^n \). The dual norm, also over vectors in \( \mathbb{R}^n \), is defined as
\[
\| z \|_* = \sup_{\| x \| = 1} | z^\top x | .
\]
For example, the dual norm of \( \| \cdot \|_\infty \) is \( \| \cdot \|_1 \). It can be seen that \( \| \cdot \|_{*\ast} = \| \cdot \| \) and that for any vectors \( x, z \),
\[
z^\top x = \| x \| \left( \frac{z^\top x}{\| x \|} \right) \leq \| z \|_* \| x \| . \tag{36}
\]
The corresponding operator norm, over \( n \times n \) real matrices, is defined as
\[
\| A \| \equiv \sup_{\| x \| = 1} \| Ax \| .
\]
For example, if \( A \) is a matrix with non-negative entries then \( \| A \|_\infty \) and \( \| A \|_1 \) can be seen to be the maximum row and column sum of \( A \), respectively. Operator norms are submultiplicative, i.e., for every operator norm \( \| \cdot \| \) and any two \( n \times n \) matrices \( A, B \),
\[
\| AB \| \leq \| A \| \| B \| . \tag{37}
\]
Finally, for any vector norm \( \| \cdot \| \), any row vector \( x^\top \) and \( n \times n \) matrix \( A \) we have that
\[
\| x^\top A \|_* \leq \| x^\top \|_* \| A \| . \tag{38}
\]

B Additional LLL Background

Recall that in the context of the General LLL, the lopsidependency graph is the directed graph on \([m]\) where each vertex \( i \) points to the vertices in \( L(i) \). Let \( G \) be an undirected graph on \([m]\) such that \( L(i) \) is a subset of the neighbors of \( i \), for every \( i \in [m] \). (One can trivially get such a \( G \) by ignoring the direction of arcs in the lopsidependency graph, but at the cost of potentially expanding the “neighborhood” of each vertex.) Given such a graph \( G \), one can restrict the sum in (4) to \( S \subseteq L(i) \) that are independent in \( G \). This is the so-called cluster expansion condition \([20]\). Whether this leads to an improvement over \([4]\), naturally, depends on how much the neighborhoods of the vertices in \( G \) are greater than their corresponding out-neighborhoods in \( L(i) \).

In fact, for a given graph \( G \) as above, the exact condition for avoiding all bad events as a function of the bounds \( \{ b_i \}_{i=1}^m \) is known, due to Shearer \([65]\). Unlike the cluster-expansion condition, though, Shearer’s condition involves a separate condition for every independent set in \( G \). Moreover, it is known that when \( \mu \) is a product measure and the dependencies between events can be expressed in terms of variable sharing (that is, under the assumptions of the variable setting \([57]\)), several works \([50, 40, 45]\) have shown that Shearer’s condition can be improved, i.e., that more permissive conditions exist.

Shearer’s condition was made constructive in the variable setting by Kolipaka and Szegedy \([50]\) and in general probability spaces by Harvey and Vondrak \([44]\) and Kolmogorov \([52]\). An important notion introduced in \([50]\) is the so-called stable set matrix of an LLL instance, which has an entry for each pair of independent sets in \( G \). The importance of the stable set matrix comes from the fact that Shearer’s condition is equivalent to the spectral radius of this matrix being strictly less than 1. We emphasize that this spectral view of Shearer’s condition is entirely different from the techniques we introduce in this paper, wherein randomized algorithms are analyzed by bounding the spectral radius of a submatrix of their transition matrix.

Finally, Scott and Sokal \([62]\) introduced a so-called soft-core LLL condition, in an effort to quantify interaction strengths between bad events (whereas in all other works two bad events either interact or they don’t). Unlike our work that quantifies general point-to-set interactions, the condition in \([62]\) only quantifies pairwise (point-to-point) interactions. Finding combinatorial applications for that condition was left as an open question in \([62]\). To the best of our knowledge, it remains open.
C  A Connection between Causality and Lopsidependency

Theorem C.1 below shows that condition (5) is the algorithmic counterpart of condition (4) in the sense that any causality graph is a lopsidependency graph with respect to the measure $\mu$ with $b_i = \gamma_i$ for all $i \in [m]$.

**Theorem C.1 ([31]).** Given a family of flaws $F = \{f_1, \ldots, f_m\}$ over a state space $\Omega$, an algorithm $A$ with causality graph $C$ with neighborhoods $\Gamma(\cdot)$ and a measure $\mu$ over $\Omega$, then for each $S \subseteq F \setminus \Gamma(i)$, we have

$$\mu(f_i \mid \bigcap_{j \in S} \overline{f_j}) \leq \gamma_i \quad (39)$$

where $\gamma_i$ are the charges of the algorithm as defined in (5).

**Proof.** Let $F_S := \bigcap_{j \in S} \overline{f_j}$. Observe that

$$\mu(f_i \mid F_S) = \frac{\mu(f_i \cap F_S)}{\mu(F_S)} = \frac{\sum_{\sigma \in f_i \cap F_S} \mu(\sigma) \sum_{\tau \in a(i, \sigma)} \rho_i(\sigma, \tau)}{\mu(F_S)} \leq \frac{\sum_{\sigma \in f_i \cap F_S} \mu(\sigma) \sum_{\tau \in F_S} \rho_i(\sigma, \tau)}{\mu(F_S)} \quad (\ast) ,$$

where the second equality holds because each $\rho_i(\sigma, \cdot)$ is a probability distribution and the third one by the definition of causality and the fact that $S \subseteq F \setminus \Gamma(i)$. Now notice that changing the order of summation in $(\ast)$ gives

$$\sum_{\tau \in F_S} \sum_{\sigma \in f_i \cap F_S} \mu(\sigma) \rho_i(\sigma, \tau) = \sum_{\tau \in F_S} \mu(\tau) \sum_{\sigma \in f_i \cap F_S} \mu(\sigma) \rho_i(\sigma, \tau) \leq \sum_{\tau \in F_S} \mu(\tau) \left( \max_{\tau' \in \Omega} \sum_{\sigma \in f_i} \frac{\mu(\sigma)}{\mu(\tau')} \rho_i(\sigma, \tau') \right) \mu(F_S) = \gamma_i .$$

$$\Box$$

D  Proof of Lemma 5.4

Our computations are similar to the ones in [53]. The following version of Chernoff Bounds will be useful:

**Lemma D.1.** Suppose $\{X_i\}_{i=1}^m \in \{0, 1\}$ are boolean variables, and set $Y_i = 1 - X_i$, $X = \sum_{i=1}^m X_i$. If $\{Y_i\}_{i=1}^m$ are negatively correlated, then for any $0 < t \leq \mathbb{E}[X]$ \n
$$\Pr[|X - \mathbb{E}[X]| > t] < 2 \exp \left(-\frac{t^2}{3\mathbb{E}[X]}\right) .$$

**Proof of part (a).** Let $v \in V$ and $\sigma \in \Omega$ be arbitrary and let $\sigma' \in \Omega$ be the (random) output (state) of $\text{RECOLOR}(v, \sigma)$. For each color $c \in \mathcal{L}_v$, let $P_v^c = \{u \in N_v : c \in R_u^v(\sigma)\}$ and define

$$\rho(c) = \sum_{u \in P_v^c} \frac{1}{|R_u^v(\sigma)| - 1} .$$
Since \( c \in R^u_w(\sigma) \) implies \( |R^u_w(\sigma)| \geq 2 \), and since \( 1 - 1/x > \exp(-1/(x - 1)) \) for \( x \geq 2 \), we see that

\[
\mathbb{E}[|L_v(\sigma')|] = 1 + \sum_{c \in \mathcal{L}_v} \prod_{u \in P_v^c} \left( 1 - \frac{1}{|R^u_w(\sigma)|} \right) > \sum_{c \in \mathcal{L}_v} \prod_{u \in P_v^c} \exp \left( -\frac{1}{|R^u_w(\sigma)| - 1} \right) = \sum_{c \in \mathcal{L}_v} e^{-\rho(c)} .
\]  
(40)

Also, since each \( R^u_w(\sigma) \) has \( |R^u_w(\sigma)| - 1 \) non-Black colors, we see that

\[
Z_v := \sum_{c \in \mathcal{L}_v} \rho(c) \leq \sum_{u \in N_v} \sum_{c \in R^u_w(\sigma)|\text{Blank}} \frac{1}{|R^u_w(\sigma)| - 1} \leq \Delta .
\]  
(41)

The fact that \( e^{-x} \) is convex implies that the right hand side of (40) is at least \( |\mathcal{L}_v| \exp(-Z_v/|\mathcal{L}_v|) \). Recalling that \( |\mathcal{L}_v| = q = (1 + \epsilon)\frac{\Delta}{\ln f} \) and combining (40) with (41) yields

\[
\mathbb{E}[|L_v(\sigma')|] > q e^{-Z_v/q} \geq (1 + \epsilon) \frac{\Delta}{\ln \sqrt{f}} e^{-\Delta/q} = 2(1 + \epsilon) \frac{\Delta}{\ln f} f^{-\frac{1}{2(1 + \epsilon)}} = 2L .
\]

Let \( X_v \) be the indicator variable that \( c \in L_v(\sigma') \) so that \( |L_v(\sigma')| = 1 + \sum_{c \in \mathcal{L}_v(\sigma')} X_v \). It is not hard to see that the variables \( Y_v = 1 - X_v \) are negatively correlated, so that applying Lemma [D.1] with \( t = \frac{1}{2}\mathbb{E}[|L_v(\sigma')|] > L \) yields

\[
\Pr \left[ |L_v(\sigma')| < \frac{1}{2}\mathbb{E}[|L_v(\sigma')|] \right] \leq 2e^{-\mathbb{E}[|L_v(\sigma')|]/12} < 2e^{-L/6} .
\]

\[\square\]

**Proof of part [b].** Let \( \Psi = \{ c \in L_v(\sigma) : \rho(c) \geq L/20 \} \setminus \text{Blank} \). The probability that \( L_v(\sigma') \) contains at least one color from \( \Psi \) is at most

\[
\mathbb{E}[|L_v(\sigma') \cap \Psi|] = \sum_{c \in \mathcal{L}_v} \prod_{u \in P_v^c} \left( 1 - \frac{1}{|R^u_w(\sigma)|} \right) \leq \sum_{c \in \mathcal{L}_v} \prod_{u \in P_v^c} \exp \left( -\frac{1}{2(|R^u_w(\sigma)| - 1)} \right) < \sum_{c \in \Psi} e^{-\rho(c)/2} ,
\]

where we used that \( c \in R^u_w(\sigma) \) implies \( |R^u_w(\sigma)| \geq 2 \), and that \( 1 - 1/x < \exp(-1/(2(x - 1))) \) for \( x \geq 2 \). Finally note that \( \sum_{c \in \Psi} e^{-\rho(c)/2} \leq q e^{-L/40} \) by the definition of the set \( \Psi \).

Recall that \( T_{v,c}(\sigma') = \{ u \in N_v : \sigma'(u) = \text{Blank} \} \) and \( c \in L_v(\sigma') \). Since \( L_v(\sigma') \subseteq R_u(\sigma') \), it follows that \( T_{v,c}(\sigma') \subseteq P_v^c \) and, therefore, \( \mathbb{E}[|T_{v,c}(\sigma')|] \leq \sum_{u \in P_v^c} 1/|R^u_w(\sigma)| \leq \rho(c) \). Since the vertices in \( P_v^c \) are colored (and thus become Blank) independently and since \( \rho(c) < L/20 \) for \( c \notin \Psi \), applying Lemma [D.1] with \( t = L/20 \) yields \( \Pr [ |T_{v,c}(\sigma')| > \mathbb{E}[|T_{v,c}(\sigma')|] + L/20 ] < 2e^{-L/60} \). Applying the union bound over all \( q \) colors, we see that the probability there is at least one \( c \notin \Psi \) for which \( |T_{v,c}(\sigma')| > L/10 \) is at most \( 2q e^{-L/60} \). Thus, with probability at least \( 1 - 3q e^{-L/60} \),

\[
\sum_{c \in L_v(\sigma') \setminus \text{Blank}} |T_{v,c}(\sigma')| = \sum_{c \in L_v(\sigma') \setminus (\Psi \cup \text{Blank})} |T_{v,c}(\sigma')| < \frac{L}{10} |L_v(\sigma')| .
\]

\[\square\]

## E Proof of Theorem 3.6

We will follow closely the approach adopted by the authors in [11]. Throughout the proof we assume that \( \epsilon \in (0, \epsilon_0) \), where \( \epsilon_0 \) is sufficiently small, and that \( f_\epsilon > 0 \) and \( \Delta_\epsilon > 0 \) are sufficiently large.

We distinguish two cases, depending on whether \( f \geq \Delta^{(2+\epsilon^2)\epsilon} \) or not. To prove Theorem 3.6 for the case \( f \geq \Delta^{(2+\epsilon^2)\epsilon} \) we will prove the following.
Theorem E.1. For every $\theta, \zeta \in (0, 1)$, there exists $\Delta_{\theta, \zeta} > 0$ such that every graph $G$ with maximum degree $\Delta \geq \Delta_{\theta, \zeta}$ in which the neighbors of every vertex span at most $\Delta^{2-(2+\zeta)\theta}$ edges, has chromatic number $\chi(G) \leq (1 + \zeta)(1 + \theta^{-1}) \frac{\Delta}{\ln \Delta}$.

Proof of Theorem E.1. We apply Theorem E.1 with $\zeta = \epsilon^2$ and $\theta = \frac{\ln f}{(2+\epsilon^2) \ln \Delta} \geq \epsilon$, so that $\Delta^2/f = \Delta^2 - (2 + \zeta)\theta$. Since $\zeta, \theta < 1$, we obtain

$$
\chi(G) \leq (1 + \zeta) \left( 1 + \frac{(2 + \zeta) \ln \Delta}{\ln f} \right) \frac{\Delta}{\ln \Delta}
$$

$$
= (1 + \zeta) \frac{\Delta}{f \ln \Delta} + (1 + \zeta)(1 + \zeta/2) \frac{\Delta}{\ln \sqrt{f}}
$$

$$
\leq (1 + 2\zeta) \frac{\Delta}{\ln \sqrt{f}} + (1 + \zeta)(1 + \zeta/2) \frac{\Delta}{\ln \sqrt{f}}
$$

$$
= \left( 2 + \frac{7\zeta}{2} + \frac{\zeta^2}{2} \right) \frac{\Delta}{\ln \sqrt{f}}
$$

$$
\leq (2 + \epsilon) \frac{\Delta}{\ln \sqrt{f}}.
$$

To prove Theorem E.1, we will perform a sequence of random halving steps, as in [11], to partition the graph into subgraphs satisfying the condition of Theorem 3.6 with $f \geq \Delta^{2-(2+\epsilon)\theta}$ and color these subgraphs using disjoint sets of colors. We will use the following lemma from [11]. As it is proven via the standard LLL, it can be made constructive using the Moser-Tardos algorithm.

Lemma E.2. Let $G(V, E)$ be a graph with maximum degree $\Delta \geq 2$ in which the neighbors of every vertex span at most $s$ edges. There exists a partition $V = V_1 \cup \ldots \cup V_k$ with $k = \Delta^{1-\theta}$, such that for every $1 \leq i \leq k$,

$$
\chi(G[V_i]) \leq (1 + \zeta)(1 + \theta^{-1}) \frac{\Delta}{\ln \Delta}.
$$

Proof of Theorem E.2. If $V_1, V_2, \ldots, V_k, k = \Delta^{1-\theta}$ is the partition promised by Lemma E.2 then

$$
\chi(G) \leq \sum_{i=1}^{\Delta^{1-\theta}} \chi(G[V_i]) \leq (1 + \zeta)(1 + \theta^{-1}) \frac{\Delta}{\ln \Delta}.
$$

To prove Theorem 3.6 for $f \in [f, \Delta^{2-(2+\epsilon)\theta}]$, we will perform a sequence of random halving steps, as in [11], to partition the graph into subgraphs satisfying the condition of Theorem 3.6 with $f \geq \Delta^{2-(2+\epsilon)\theta}$ and color these subgraphs using disjoint sets of colors. To perform the partition we use the following lemma from [11]. As it is proven via the standard LLL, it can be made constructive using the Moser-Tardos algorithm.

Lemma E.3 ([11]). Let $G(V, E)$ be a graph with maximum degree $\Delta \geq 2$ in which the neighbors of every vertex span at most $s$ edges. There exists a partition $V = V_1 \cup V_2$ such that the induced subgraph $G[V_i], i = 1, 2$, has maximum degree at most $\Delta/2 + 2\sqrt{\Delta \ln \Delta}$ and the neighborhoods of every vertex in $G[V_i], i = 1, 2$, span at most $s/4 + 2\Delta^{1/2} \sqrt{\ln \Delta}$ edges.

We will also use the following lemma whose proof, presented in Section E.2, is almost identical to a similar statement in the proof of Theorem 1.1 of [11].
Lemma E.4. Given $\Delta, f$ sufficiently large, let the sequences $\Delta_t$ and $s_t$ be defined as follows. $\Delta_0 = \Delta, s_0 = \Delta^2/f$ and

$$\Delta_{t+1} = \Delta_t/2 + 2\sqrt{\Delta_t \ln \Delta_t}, \quad s_{t+1} = s_t/4 + 2\Delta_t^2 \sqrt{\ln \Delta_t}.$$  

For any $\delta \in (0, 1/100)$ and $\zeta > 0$ such that $\zeta(2 + \delta) < 1/10$, let $j$ be the smallest integer for which $f > \left(\frac{(1+\delta)\Delta}{2}\right)^{(2+\delta)\zeta}$. Then $\Delta_j \leq (1+\delta)\Delta/2^j$ and $s_j \leq \left(1+\delta\right)^2/2^j/f$.

Proof of Theorem 3.6 for $f \in [\epsilon f, \Delta^{(2+\epsilon^2)}]$. Let $\epsilon_0 = 1/11$. For $\epsilon \in (0, \epsilon_0]$, let $\delta = \zeta = \epsilon^2$. Since $\zeta(2 + \delta) < 1/10$, apply Lemma E.4 and let $j = j(\Delta, f, \delta, \zeta)$ be the integer described therein. Let $S$ be the process which, given a graph $G$, does nothing if $\Delta(G) < 2$, and otherwise partitions $G$ as described in Lemma E.3. Apply $S$ to $G$ to get subgraphs $G[V_1], G[V_2]$. Apply $S$ to $G[V_1], G[V_2]$ to get $G[V_{1,1}], G[V_{1,2}], G[V_{2,1}], G[V_{2,2}]$. And so on, $j$ times, obtaining a partition of $G$ into at most $2^j$ induced subgraphs. Observe that for each such subgraph $H$, either $\Delta(H) < 2$ and, thus, $\chi(H) \leq 2$ or, by Lemma E.4, $\Delta(H) \leq (1+\delta)\Delta/2^j =: \Delta_*$ and the neighbors of every vertex in $H$ span at most $\Delta_*^2/f$ edges, where $f \geq \Delta_*^{(2+\delta)\zeta} = \Delta_*^{(2+\epsilon^2)\zeta} \geq \Delta_*^{(2+\epsilon^2)\zeta}$. Therefore, by the already established case of Theorem 3.6, either $\chi(H) \leq 2$, or $\chi(H) \leq (2+\zeta)\Delta_*/\ln f$. Thus,

$$\chi(G) \leq 2^j \max\left\{2, (2+\zeta)\frac{(1+\delta)\Delta/2^j}{\ln f}\right\} \leq \max\left\{2^{j+1}, (2+\zeta)\frac{(1+\delta)\Delta}{\ln \sqrt{f}}\right\}. $$  

To bound $2^{j+1}$ from above we first observe that for all $f$ sufficiently large, i.e., for all $f \geq f_\epsilon$,

$$\left(\frac{(1+\delta)\Delta}{2\ln \sqrt{f}}\right)^{(2+\delta)\zeta} = \left(2(1+\delta)\ln \sqrt{f}\right)^{(2+\delta)\zeta} < f.$$

Now, since $j$ was defined as the smallest integer for which $\left(\frac{(1+\delta)\Delta}{2\ln \sqrt{f}}\right)^{(2+\delta)\zeta} < f$, we see that (42) implies $2^j < \frac{\Delta}{2\ln \sqrt{f}}$ and, therefore, $2^{j+1} \leq \frac{\Delta}{\ln \sqrt{f}}$. Finally, we observe that $(2+\zeta)(1+\delta) = (2+\epsilon^2)(1+\epsilon^2) < 2+\epsilon$ for all $\epsilon \in (0, \epsilon_0]$. Therefore, as claimed,

$$\chi(G) \leq \left(2+\epsilon\right)\frac{\Delta}{\ln \sqrt{f}}.$$  

E.1 Proof of Lemma E.2

We follow an approach similar to the one of Lemma 2.3 in [11] making appropriate modifications as needed. First we partition the vertices of $G$ into $\Delta^{1-\theta}$ parts by coloring them randomly and independently with $\Delta^{1-\theta}$ colors. For a vertex $v$ and a neighbor $u$ adjacent to it, call $u$ a bad neighbor of $v$, if $u$ and $v$ have at least $\Delta^{1-{(1+\zeta/2)}^\theta}$ common neighbors. Otherwise, say that $u$ is a good neighbor. Since the neighbors of every vertex span at most $\Delta^{2-{(2+\zeta)}^\theta}$ edges, there are at most $2\Delta^{1-{(1+\zeta/2)}^\theta}$ bad neighbors for any vertex in $G$.

For any vertex $v$, define three types of bad event with respect to the random partitioning experiment.

- $A_v$: more than $(1+\theta)\Delta^\theta$ neighbors of $v$ receive the same color as $v$.
- $B_v$: more than $\frac{\Delta}{\theta\zeta}$ bad neighbors of $v$ receive the same color as $v$.
- $C_v$: the good neighbors of $v$ that receive the same color as $v$ span more than $\frac{100}{(\theta\zeta)^2}$ edges.
We will use the symmetric version of the Local Lemma \cite{28} to show that we can find a coloring of the graph that avoids all bad events. First, note that each of the bad events $A_v, B_v, C_v$ is independent of all but at most $\Delta^2$ others, as it independent of all events $A_u, B_u, C_u$ corresponding to vertices $u$ whose distance from $v$ is more than 2. Since the degree of any vertex in its colors class is binomially distributed with mean at most $\Delta^\theta$, standard Chernoff estimates imply that the probability that $v$ has more than $(1 + \theta)\Delta^\theta$ neighbors of the same color as that of $v$ is at most $e^{-\Omega(\Delta^\theta)}$, which means that $\Pr[A_v] < \Delta^{-3}$ for large enough $\Delta$. Moreover, we also have

$$\Pr[B_v] \leq \left(\frac{2\Delta-1-(1+\zeta/2)\theta}{\theta \zeta} \right) \left(\frac{1}{\Delta^{1-\theta}}\right)^{\frac{9}{\theta \zeta}} \leq \left(\frac{2}{\Delta^{1-\theta}}\right)^{\frac{9}{\theta \zeta}} \leq \Delta^{-3},$$

for large enough $\Delta$. Finally, to bound the probability of $C_v$ we make the following observation. If a graph has at least $e^2$ edges, then either it has a vertex of degree at least $e$, or every vertex has degree strictly less than $e$, implying that the graph can be edge-colored with $\lceil e \rceil$ colors, in which case the largest color class must contain at least $e^2/\lceil e \rceil \geq e - 1$ edges. Thus, a graph with more than $100/(\theta\zeta)^2$ edges either has a vertex of degree at least $10/(\theta\zeta) \geq 9/(\theta\zeta)$ or a matching with at least $10/(\theta\zeta) - 1 \geq 9/(\theta\zeta)$ edges, where the inequality follows from the fact that $\theta, \zeta < 1$. Thus, $C_v$ can happen only if there is a good neighbor $v$ of $v$ such that $u$ and $v$ have at least $9/(\theta\zeta)$ common neighbors with the same color as $v$, or if there is a matching of size at least $9/(\theta\zeta)$ on the good neighbors of $v$ that have the same color as $v$. The probabilities of the first and second of these events are bounded, respectively, by

$$\Delta \left(\frac{\Delta-1-(1+\zeta/2)\theta}{\theta \zeta} \right) \left(\frac{1}{\Delta^{1-\theta}}\right)^{\frac{9}{\theta \zeta}} \leq \left(\frac{1}{\Delta^{1-\theta}}\right)^{\frac{9}{\theta \zeta}} \leq \frac{1}{2} \Delta^{-3},$$

and

$$\left(\frac{\Delta-2-(2+\zeta)\theta}{\theta \zeta} \right) \left(\frac{1}{\Delta^{1-\theta}}\right)^{\frac{9}{\theta \zeta}} \leq \left(\frac{1}{\Delta^{1-\theta}}\right)^{\frac{9}{\theta \zeta}} \leq \frac{1}{2} \Delta^{-3}. $$

Therefore the probability of $C_v$ is at most $\Delta^{-3}$. Thus, the Local Lemma applies since each bad event has probability at most $\Delta^{-3}$ and is independent of all but at most $\Delta^2$ other bad events. This means that we can find a partition $V = V_1, \ldots, V_k$, where $k = \Delta^{1-\theta}$, so that in each induced subgraph $G[V_i]$, every vertex: has degree at most $(1 + \theta)\Delta^\theta$, has at most $\frac{10}{\theta \zeta}$ bad neighbors, and is contained in at most $\frac{100}{(\theta \zeta)^2}$ triangles in which both other vertices are good. We will show that, given such a partition, each $G[V_i]$ can be colored with at most $\frac{(1+\zeta)/(1+\theta^{-1})\Delta^\theta}{\ln \Delta}$ colors, assuming $\Delta$ is large enough.

To see this, consider the partition $B_i, V_i \setminus B_i$ of $V_i$, where $B_i$ is the set of vertices $u \in V_i$ for which there exists a vertex $v \in V_i$, such that $u$ is a bad neighbor of $v$. We claim that $\chi(G[B_i]) \leq \frac{20}{\theta \zeta} + 1$ and $\chi(G[V_i \setminus B_i]) \leq \frac{(1+\zeta/2)(1+\theta^{-1})\Delta^\theta}{\ln \Delta}$. Assuming this claim, observe that

$$\chi(G[V_i]) \leq \frac{20}{\theta \zeta} + 1 + \frac{(1+\zeta/2)(1+\theta^{-1})\Delta^\theta}{\ln \Delta} \leq \frac{(1+\zeta)(1+\theta^{-1})\Delta^\theta}{\ln \Delta},$$

where the last inequality holds for all $\Delta \geq \Delta_{\theta, \zeta}$.

To see the first part of the claim, note that it is well-known (and easy to see) that if a graph has an orientation with maximum outdegree $d$, then it is $(2d + 1)$-colorable. Consider the orientation of the graph on $B_i$ that results when every vertex points to its bad neighbors in $B_i$. Clearly, the maximum outdegree is at most $\frac{10}{\theta \zeta}$ and, thus, $\chi(G[B_i]) \leq \frac{20}{\theta \zeta} + 1$. 

38
To see the second part of the claim, observe that each vertex of $G[V_i \setminus B_i]$ is contained in at most $\frac{100}{(\theta \zeta)^2}$ triangles. Let $\Delta_\ast = (1 + \theta)\Delta^\theta$ and

$$f = \frac{((1 + \theta)\Delta^\theta)^2}{100/(\theta \zeta)^2} = \frac{(\theta \zeta)^2 \Delta^2_\ast}{100} \geq \Delta^2_\ast \left(\frac{\ln \Delta_\ast}{\ln \Delta}\right)^2,$$

where the last inequality holds for $\Delta \geq \Delta_{\theta, \zeta}$. Applying Theorem 3.5 to $G[V_i \setminus B_i]$ (by plugging in $\zeta/3$ for the $\epsilon$ in Theorem 3.5) we get that, for all $\Delta \geq \Delta_{\theta, \zeta}$,

$$\chi_\ell(G[V_i \setminus B_i]) \leq (1 + \zeta/3) \frac{\Delta_\ast}{\ln \sqrt{f}} \leq (1 + \zeta/3) \frac{(1 + \theta)\Delta^\theta}{\ln \Delta + \ln \frac{(1 + \theta)\theta \zeta}{10}} \leq (1 + \zeta/2) \frac{(1 + \theta)\Delta^\theta}{\theta \ln \Delta} \leq (1 + \zeta/2) \frac{(1 + \theta^{-1})\Delta^\theta}{\ln \Delta},$$

as claimed.

E.2 Proof of Lemma E.4

Let $\epsilon' := \zeta(2 + \delta)$ and recall that $\epsilon' < \frac{1}{10}$ by hypothesis. By the definition of $j$, for every $t < j$,

$$\Delta_t \geq \Delta / (2^t) > \frac{f^{\frac{1}{2^t}}}{1 + \delta},$$

and $f^{\frac{1}{2^t}}/(1 + \delta)$ can be made arbitrarily large by taking $f$ to be sufficiently large. Hence, we can assume that $\Delta_t$ is sufficiently large in order for $\Delta_{t+1} \leq \Delta_t^\frac{1}{2^t} + \Delta_t^\frac{1}{2^t}$ to hold. Taking cube roots and subtracting $\frac{1}{2^{t+1}}$ from both sides we get

$$\Delta_t^\frac{1}{2^{t+1}} - \frac{1}{2^{t+1}} \leq \frac{1}{2^t}(\Delta_t^\frac{1}{2^t} + 1) - \frac{1}{2^t} = \frac{1}{2^t} \left(\Delta_t^\frac{1}{2^t} - \frac{1}{2^t - 1}\right).$$

Therefore,

$$\Delta_j^\frac{1}{2^j} - \frac{1}{2^j - 1} \leq \frac{1}{2^{j/3}} \left(\Delta_0^\frac{1}{2^{j/3}} - \frac{1}{2^{j/3} - 1}\right).$$

(43)

Since $\Delta_0 = \Delta, 2^{j/3} - 1 > \frac{1}{4}$ and $\Delta/2^{j-1} > \frac{f^{\frac{1}{2^j}}}{1 + \delta}$ is large enough, (43) implies that

$$\Delta_j^\frac{1}{2^j} \leq \Delta^{\frac{1}{2^{j/3}}} + 4 \leq (1 + \delta)^{\frac{1}{2^{j/3}}} \Delta^{\frac{1}{2^{j/3}}}.$$

39
Therefore, we have shown that $\Delta_j \leq (1 + \delta) \frac{\Delta}{2^j}$. Note also that the same proof shows that for every $t \leq j$ we have that $\Delta_t \leq (1 + \delta) \frac{\Delta}{2^t}$.

We turn now to the claim regarding $s_j$. For all $t < j$, we have by definition

$$s_t \geq \frac{s_0}{4^t} = \frac{\Delta^2}{4^t f} = \frac{1}{(1 + \delta)^2} \left( \frac{(1 + \delta) \Delta}{2^t} \right)^2 \geq \frac{1}{(1 + \delta)^2} \left( \frac{(1 + \delta) \Delta}{2^t} \right)^{2 - \epsilon'} \geq \frac{1}{(1 + \delta)^2} \Delta_t^{2 - \epsilon'}, \quad (44)$$

where in the last inequality we used the fact that $\Delta_t \leq (1 + \delta) \frac{\Delta}{2^t}$ for all $t \leq j$. Using (44) to bound $\Delta_t$ in the expression that defines $s_t$, we get

$$s_{t+1} \leq \frac{s_t}{4} + 2((1 + \delta)^2 s_t)^{\frac{3}{2} - \epsilon' \frac{1}{2}}. \quad (45)$$

To bound the r.h.s. of (45) we recall that $\epsilon' < \frac{1}{10}$ implying $\frac{3}{2(2 - 2\epsilon')} < \frac{3}{2(2 - 1/5)} = \frac{5}{6}$. Assuming that $f$ (and, thus, $\Delta_t$) is large enough, we obtain

$$s_{t+1} \leq \frac{s_t}{4} + 3s_t^{5/6} \leq \frac{1}{4}(s_t^{1/6} + 2)^6 = \frac{1}{4}(s_t + 12s_t^{5/6} + 60s_t^{2/3} + \cdots). \quad (46)$$

Hence, taking 6-th roots and subtracting $\frac{5}{6} \frac{1}{6 - 1}$ from both sides, we obtain

$$s_{t+1}^{1/6} - \frac{5}{61/6 - 1} \leq \frac{1}{4\sqrt[6]{6}}(s_t^{1/6} + 2) - \frac{5}{61/6 - 1} \leq \frac{1}{4\sqrt[6]{6}} \left( s_t^{1/6} - \frac{5}{61/6 - 1} \right).$$

Therefore

$$s_j^{1/6} - \frac{5}{61/6 - 1} \leq \frac{1}{4\sqrt[6]{6}} \left( s_0^{1/6} - \frac{5}{61/6 - 1} \right),$$

and, since $s_0 = \Delta^2 / f$,

$$s_j^{1/6} \leq \frac{\Delta^2}{4^j f 1/6} + \frac{5}{61/6 - 1} \leq \left( \frac{\Delta^2}{4^j f} \right)^{1/6} + 15.$$  

Since

$$\frac{\Delta^2}{4^j f} = \left( \frac{\Delta}{2^j} \right)^2 \frac{1}{f} = \left( \frac{\Delta}{2^j-1} \right)^2 \frac{1}{4^j f} = \left( \frac{1}{1 + \delta} \right)^2 \frac{1}{4^j f} = \frac{f^{2^j - 1}}{4(1 + \delta)^2}$$

can be made arbitrarily large by taking $f$ sufficiently large, we see that

$$\left( \frac{\Delta^2}{4^j f} \right)^{1/6} + 15 \leq (1 + \delta)^{1/3} \left( \frac{\Delta^2}{4^j f} \right)^{1/6}.$$  

Thus, $s_j \leq ((1 + \delta) \Delta/2^j)^2 / f$, completing the proof.

**F Proof of Proposition 3.1**

We use the term “with high probability” to refer to probabilities that tend to 1 as $n$ goes to infinity. Corollary 3.1 follows in a straightforward way from the following lemma.
Lemma F.1. For any \( \delta \in (0,1) \) there exists a constant \( d_0 \) such that, for any \( d \in (d_0 \ln n, (n \ln n)^{\frac{1}{3}}) \), each vertex of the random graph \( G = G(n, d/n) \) is contained in at most \( \Delta^\delta \) triangles with high probability, where \( \Delta \) is the maximum degree of \( G \).

Proof of Corollary 3.1. According to [8], for a graph \( G \in G(n, d/n) \) we know that with high probability
\[
\chi(G) = \frac{1}{2} d \ln d (1 + o(1)) \tag{47}
\]
for large enough \( d \). Thus, we can apply Theorem 3.5 with parameter \( \zeta > 0 \) since
\[
f = \frac{\Delta^2}{\Delta^\delta} > \frac{\Delta^2}{\Delta^{\frac{1}{3}}} (\ln \Delta)^2,
\]
for large enough \( \Delta \). This yields an upper bound \( q \) on the chromatic number of \( G \) that is at most
\[
q = \left(1 + \frac{\zeta}{\ln \sqrt{f}}\right) \frac{\Delta}{\ln \sqrt{f}} \leq \left(1 + \frac{\zeta}{\ln \sqrt{f}}\right) \frac{\Delta}{\ln \sqrt{f}} \leq \left(1 + \frac{1}{2}\right) \frac{\Delta}{\ln \sqrt{f}} \tag{48}
\]
Moreover, since the expected degree of every vertex of \( G \) is \( d \) and its distribution is binomial with parameter \( \frac{d}{n} \), standard Chernoff bounds and the union bound imply that for any \( \eta \in (0,1) \), \( \Delta \leq (1 + \eta)d \) with high probability, for large enough \( d_0 \).

Combining the latter fact with (47) and (48), we deduce that we can find an arbitrarily small constant \( \eta' \in (0,1) \) such that
\[
q \leq (2 + \eta') \chi(G)
\]
by choosing \( \zeta \) and \( \eta \) sufficiently small. Picking \( \eta' = \frac{\eta}{1-\eta} \) we obtain \( \chi(G) \geq \frac{d}{2+\eta} \geq q(\frac{1}{2} - \epsilon) \), concluding the proof of Proposition 3.1.

\( \square \)

F.1 Proof of Lemma F.1

Let \( \Delta_v \) be the random variable that equals the degree of vertex \( v \) of \( G \). Observe that \( \Delta_v \sim \text{Binom}(n-1, \frac{d}{n}) \) and, therefore, using a standard Chernoff bound and the fact that \( d \geq d_0 \log n \) we get that
\[
\Pr \left[ \Delta_v \notin (1 \pm \frac{1}{10})d \right] \leq \frac{1}{n^2}
\]
for large enough \( d_0 \). Thus, by a union bound we get that \( \Pr[\Delta \notin (1 \pm \frac{1}{10})d] \leq \frac{1}{n} \).

Let \( T_v \) be the number of triangles that contain vertex \( v \) and \( B \) be the event that \( \Delta \notin (1 \pm \frac{1}{10})d \). Then,
\[
\Pr[T_v > \Delta^\delta] \leq \Pr[T_v > \Delta^\delta | B] + \Pr[B] \leq \frac{\Pr[(T_v > \Delta^\delta) \cap B]}{1 - \frac{1}{n}} + \frac{1}{n}.
\]
Observe that $T_v \sim \text{Binom}\left(\binom{n-1}{2}, \left(\frac{d}{n}\right)^3\right)$ and $\mathbb{E}[T_v] \leq \frac{d^3}{2n}$. Thus, for any fixed value of $\Delta \in (1 \pm \frac{1}{10})d$, setting $1 + \beta = \frac{\Delta d}{d^3/2n}$ and using a standard Chernoff bound we obtain:

$$\Pr[T_v > \Delta^\delta] \leq e^{-\frac{\beta^2 d^3/2n}{3}} \leq \frac{1}{n^2}$$

since

$$\beta \geq \frac{\left((1 - \frac{1}{10})d\right)^\delta - d^3/2n}{d^3/2n} > 0,$$

$$\frac{1}{3} \beta^2 \frac{d^3}{2n} \geq \frac{1}{3} \left(\frac{\left((1 - \frac{1}{10})d\right)^\delta - d^3/2n}{d^3/2n}\right)^2 \geq 2 \ln n,$$

whenever $d \in [d_0 \ln n, (n \ln n)^\frac{1}{3}]$ and for large enough $n$ and $d_0$. Taking a union bound over $v$ concludes the proof of the lemma.