We give an algorithmic local lemma by establishing a sufficient condition for the uniform random walk on a directed graph to reach a sink quickly. Our work is inspired by Moser’s entropic method proof of the Lovász Local Lemma (LLL) for satisfiability and completely bypasses the Probabilistic Method formulation of the LLL. In particular, our method works when the underlying state space is entirely unstructured. Similarly to Moser’s argument, the key point is that the inevitability of reaching a sink is established by bounding the entropy of the walk as a function of time.
Lemma (LLL), asserting that such positivity also holds under a condition of limited dependence among the events in $\mathcal{A}$. The idea of the Local Lemma was first circulated by Lovász in the early 1970s in an unpublished note. It was published in Erdős and Lovász [1975]. The general form below is also due in unpublished form to Lovász and was given in Spencer [1977].

**General LLL.** Let $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$ be a set of events and let $D(i) \subseteq (m \setminus \{i\})$ denote the set of indices of the dependency set of $A_i$, that is, $A_i$ is mutually independent of all events in $\mathcal{A} \setminus \{A_i \cup \bigcup_{j \in D(i)} A_j\}$. If there exist finite, nonnegative real numbers $\{\mu_i\}$ such that, for all $i \in [m]$,

$$
\Pr(A_i) \prod_{j \in \{i\} \cup D(i)} (1 + \mu_j) \leq \mu_i,
$$

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

**Remark 1.** Condition (1) is equivalent to $\Pr(A_i) \leq x_i \prod_{j \in D(i)} (1 - x_j)$, where $x_i = \mu_i/(1 + \mu_i)$. As we will see the formulation (1) facilitates comparisons.

Erdős and Spencer [1991] noted that one can replace the LLL’s requirement that each bad event is dependent with few other bad events with the weaker requirement that each bad event is *negatively correlated* with few other bad events. That is, for each bad event $A_i$ there should only be few other bad events whose nonoccurrence may boost $A_i$’s probability of occurring; the nonoccurrence of any subset of the remaining events should leave $A_i$ either unaffected, or make it less likely. A natural setting for this, so-called lopsided LLL arises when one seeks a collection of permutations satisfying a set of constraints and considers the uniform measure on them. While the bad events (constraint violations) are now typically densely dependent (as fixing the image of even just one element affects all others), one can often establish sufficient negative correlation among the bad events to apply the lopsided LLL.

**Lopsided LLL.** Erdős and Spencer [1991]. Let $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$ be a set of $m$ events. For each $i \in [m]$, let $\Gamma(i)$ be a subset of $[m] \setminus \{i\}$ such that $\Pr(A_i \mid \bigcap_{j \in \Gamma(i)} A_j) \leq \Pr(A_i)$ for every $S \subseteq [m] \setminus (\Gamma(i) \cup \{i\})$. If there exist finite, nonnegative real numbers $\{\mu_i\}$ such that, for all $i \in [m]$,

$$
\Pr(A_i) \prod_{j \in \{i\} \cup \Gamma(i)} (1 + \mu_j) \leq \mu_i,
$$

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

In the context of the general LLL it is natural to define the *dependence digraph* $D$ of a collection of events $\{A_1, A_2, \ldots, A_m\}$ as having a vertex $v_i$ for each event $A_i$ and an arc $(v_i, v_j)$ if and only if $j \in D(i)$, noting that there exist systems of events such that $D$ contains an arc $(i, j)$ but not arc $(j, i)$. The lopsided dependence digraph is the sparsification $D_L$ of $D$ wherein each event $A_i$ points only to the events negatively correlated with it, that is, the elements of the set $\Gamma(i) \subseteq D(i)$. Let $G$ be the undirected graph that results by ignoring arc direction in $D_L$. Also, observe that condition (2) can be trivially rewritten (expanded) as

$$
\Pr(A_i) \sum_{S \subseteq \{i\} \cup \Gamma(i)} \prod_{j \in S} \mu_j \leq \mu_i.
$$

Relatively recently, Bissacot et al. [2011] improved the lopsided LLL when the graph $G$ is not triangle-free. Specifically, they showed that the conclusion of the lopsided LLL remains valid if the summation in Equation (3) is restricted to those sets $S \subseteq \{i\} \cup \Gamma(i)$ that are independent in $G$. 

Journal of the ACM, Vol. 63, No. 3, Article 22, Publication date: July 2016.
1.1. Constructive Versions

As one can imagine, after proving that \( \Omega \) contains flawless objects via the LLL, it is natural to ask if some flawless object can be found efficiently. Making the LLL constructive has been a long quest, starting with the work of Beck [1991], with subsequent works of Alon [1991], Molloy and Reed [1999], Czumaj and Scheideler [2000], Srinivasan [2008], and others. Each work established a method for finding flawless objects efficiently, but in all cases under significant additional conditions relative to the LLL. The breakthrough was made by Moser [2009] who showed that a shockingly simple algorithm nearly matches the LLL condition for \( k \)-CNF formulas. Very shortly afterwards, Moser and Tardos in a landmark paper [Moser and Tardos 2010] made the general LLL constructive for all product measures over explicitly presented variables.

Specifically, in the so-called variable setting of Moser and Tardos [2010], each event \( A_i \) is associated with the set of variables \( \text{vbl}(A_i) \) that determine it so that \( j \in D(i) \) if and only if \( \text{vbl}(A_i) \cap \text{vbl}(A_j) \neq \emptyset \). Moser and Tardos [2010] proved that if the condition (1) of the general LLL holds, then repeatedly selecting any occurring event \( A_i \) (flaw present) and resampling every variable in \( \text{vbl}(A_i) \) independently of all others leads to an elementary event in which no event in \( A \) holds (flawless object) after a polynomial number of resamplings. Pegden [2014] strengthened the result of Moser and Tardos [2010] by showing that its conclusion still holds if the condition (1) of the general LLL is replaced by Equation (3), in which the summation is restricted to independent sets, that is, under the condition of Bissacot et al. [2011] mentioned earlier. Kolipaka and Szegedy [2011] showed that the algorithm of Moser and Tardos [2010], in fact, converges in polynomial time under the criterion of Shearer [1985], the most generous condition under which \( \Pr[\cap_i \overline{A_i}] > 0 \) for symmetric dependency graphs. As the criterion of Shearer is not efficiently verifiable, Kolipaka et al. [2012] gave a series of intermediate conditions, between the general LLL and Shearer’s criterion, for the algorithm of Moser and Tardos [2010] to terminate, most notably the efficiently verifiable Clique LLL. On the other hand, with the notable exception of CNF-SAT, none of these results applies to the lopsided LLL, which remained nonconstructive.

Very recently, Harris and Srinivasan [2014] made the lopsided LLL constructive for the uniform measure on Cartesian products of permutations. Among other results, this yielded an efficient algorithm for constructing \( n \times n \) Latin Squares when each color appears at most \( \Delta \leq (27/256)n \) times, matching the best nonconstructive bound due to Bissacot et al. [2011] (who improved the original \( \Delta \leq n/(4e) \) bound of Erdős and Spencer [1991] by exploiting the local density of the lopsided dependency graph). Harris and Srinivasan [2014] pointed out that, while the permutation setting is the most common-use case, the lopsided LLL has been gainfully applied to other settings [Lu et al. 2013; Mohr 2013] including hypergraph matchings, set partitions, and spanning trees, and asked if their results can be extended beyond permutations. In particular, they left as a canonical open problem whether the results of Dudek et al. [2012] regarding Hamilton Cycles in edge-colored hypergraphs can be made constructive.

After a preliminary version [Achlioptas and Iliopoulos 2014] of our work appeared, Harvey and Vondrák [2015] made the lopsided LLL algorithmic for any measure, given resampling oracles that respect the lopsidedependency graph. Their result is incomparable to our work as the two settings are subject to different restrictions. See Section 4.

2. A NEW FRAMEWORK

Inspired by the breakthrough of Moser [2009], we take a more direct approach to finding flawless objects, bypassing the probabilistic formulation of the existence question. Specifically, we replace the measure on \( \Omega \) by a directed graph \( D \) on \( \Omega \) and we seek...
flawless objects by taking random walks on $D$. With this in mind, we refer to the elements of $\Omega$ as states. As in Moser [2009], each state transformation (step of the walk) $\sigma \rightarrow \tau$ will be taken to address a flaw present at $\sigma$. Naturally, a step may eradicate other flaws beyond the one addressed, but may also introduce new flaws (and, in fact, may fail to eradicate the addressed flaw). By replacing the measure with a directed graph we achieve two main effects:

—Both the set of objects $\Omega$ and every flaw $f \subseteq \Omega$ can be entirely amorphous. That is, $\Omega$ does not need to have product form $\Omega = D_1 \times \cdots \times D_n$, as in the work of Moser and Tardos [2010], or any form of symmetry, as in the work of Harris and Srinivasan [2014]. For example, $\Omega$ can be the set of all Hamiltonian cycles of a graph, a set of very high complexity.

—The set of transformations for addressing a flaw $f$ can differ arbitrarily among the different states $\sigma \in f$, allowing the actions to adapt to the “environment.” This is in sharp contrast with all past algorithmic versions of the LLL, in which either no or very minimal adaptivity was possible. As we discuss in Section 4, this moves the Local Lemma from the Probabilistic Method squarely within the purview of Algorithm Design.

Concretely, for each $\sigma \in \Omega$, let $U(\sigma) = \{ f \in F : \sigma \in f \}$, that is, $U(\sigma)$ is the set of flaws present in $\sigma$. For each $\sigma \in \Omega$ and $f \in U(\sigma)$ we require a set $A(f, \sigma) \subseteq \Omega$ that must contain at least one element other than $\sigma$, which we refer to as the set of possible actions for addressing flaw $f$ in state $\sigma$. To address flaw $f$ in state $\sigma$ we select uniformly at random an element $\tau \in A(f, \sigma)$ and walk to state $\tau$, noting that possibly $\tau = \sigma \in A(f, \sigma)$. Our main point of departure is that now the set of actions for addressing a flaw $f$ in each state $\sigma$ can depend arbitrarily on the state, $\sigma$, itself.

We represent the set of all possible state transformations as a multidigraph $D$ on $\Omega$, formed as follows: for each state $\sigma$, for each flaw $f \in U(\sigma)$, for each state $\tau \in A(f, \sigma)$ place an arc $\sigma \xrightarrow{f} \tau$ in $D$, that is, an arc labeled by the flaw being addressed. Thus, $D$ may contain pairs of states $\sigma, \tau$ with multiple $\sigma \rightarrow \tau$ arcs, each such arc labeled by a different flaw, each such flaw $f$ having the property that moving to $\tau$ is one of the actions for addressing $f$ at $\sigma$, that is, $\tau \in A(f, \sigma)$. Since we require that the set $A(f, \sigma)$ contains at least one element other than $\sigma$ for every flaw in $U(\sigma)$, we see that a vertex of $D$ is a sink if and only if it is flawless.

We focus on digraphs satisfying the following condition.

Atomicity. $D$ is atomic if, for every flaw $f$ and state $\tau$, there is at most one arc incoming to $\tau$ labeled by $f$.

The purpose of atomicity is to capture “accountability of action.” In particular, note that if $D$ is atomic, then every walk on $D$ can be reconstructed from its final state and the sequence of labels on the arcs traversed, as atomicity allows one to unambiguously trace back the walk. To our pleasant surprise, in all applications we have considered so far, we have found atomicity to be “a feature, not a bug,” serving as a very valuable aid in the design of flaws and actions, that is, of algorithms. A fruitful way to think about atomicity is to consider the case in which $\Omega$ and $F$ have product structure over a set of variables, for example, a Constraint Satisfaction Problem. In that case, the following suffice to imply atomicity:

1. Each constraint (flaw) forbids exactly one joint value assignment to its underlying variables.
2. Each state transition modifies only the variables of the violated constraint (flaw) that it addresses.
Condition 1 expresses a purely syntactic requirement: compound constraints must be broken down to constituent parts akin to satisfiability constraints. So, for example, to encode graph $q$-colorability we must write $q$ constraints (flaws) per edge, one for each color. In most settings, not only is such a breakdown straightforward, but is also advantageous, as it affords a more refined accounting of conflict between constraints. Condition 2, on the other hand, is a genuine restriction reflecting the idea of “focusing” introduced by Papadimitriou [1991], that is, that every state transformation should be the result of attempting to eradicate some specific flaw.

To see that Conditions 1 and 2 imply atomicity, imagine that there exist arcs $\sigma_1 \xrightarrow{f} \tau$ and $\sigma_2 \xrightarrow{f} \tau$, that is, two state transformations addressing the same flaw $f$ leading to the same state $\tau$. Since $f$ must be present in both $\sigma_1$ and $\sigma_2$, Condition 1 implies that if $\sigma_1 \neq \sigma_2$, then there exists at least one variable $v$ not bound by $f$ that takes different values in $\sigma_1, \sigma_2$. In that case, though, Condition 2 implies that $v$ will have the same value before and after each of the two transformations, leading to a contradiction.

Having defined the multidigraph $D$ on $\Omega$, we will now define a digraph $C$ on the set of flaws $F$, reflecting some of the structure of $D$.

Potential Causality. For each arc $\sigma \xrightarrow{f} \tau$ in $D$ and each flaw $g$ present in $\tau$, we say that $f$ causes $g$ if $g = f$ or $g \not\in \sigma$. If $D$ contains any arc in which $f$ causes $g$, we say that $f$ potentially causes $g$.

Potential Causality Digraph. The digraph $C = C(\Omega, F, D)$ of the potential causality relation, that is, the digraph on $F$ where $f \rightarrow g$ if and only if $f$ potentially causes $g$, is called the potential causality digraph. The neighborhood of a flaw $f$ is $\Gamma(f) = \{g : f \rightarrow g \text{ exists in } C\}$.

In the interest of brevity, we will call $C$ the causality digraph, instead of the potential causality digraph. It is important to note that $C$ contains an arc $f \rightarrow g$ if there exists even one state transition aimed at addressing $f$ that causes $g$ to appear in the new state. In that sense, $C$ is a “pessimistic” estimator of causality (or, alternatively, a lossy compression of $D$). This pessimism is both the strength and the weakness of our approach. On the one hand, it makes it possible to extract results about algorithmic progress without tracking the evolution of state. On the other hand, it gives good results only when $C$ can remain sparse even in the presence of such stringent arc inclusion. We feel that this tension is meaningful: maintaining the sparsity of $C$ requires that the actions for addressing each flaw across different states are coherent with respect to the flaws that they cause.

Without loss of generality (and to avoid certain trivialities), we can assume that $C$ is strongly connected, implying that $\Gamma(f) \geq 1$ for every $f \in F$. To see this, let $C_1, \ldots, C_h$ be the strongly connected components of $C$ and consider the DAG with vertices $c_1, \ldots, c_h$, where for $i \neq j$, $c_i$ points to $c_j$ if and only if there exist $f \in C_i$ and $g \in C_j$ such that $f \rightarrow g$ exists in $C$. If we have a sufficient condition for finding flawless objects when the causality digraph is strongly connected, then we can take any source vertex $c_i$ in the DAG and repeatedly address flaws in $C_i$ until we reach a state $\sigma_i \in \Omega$ that is $C_i$-flawless, at which point we remove $c_i$ from the DAG. If $\sigma_i$ has other flaws, we select a new source vertex $c_j$ and repeat the same idea continuing from $\sigma_i$. The actions that will be taken to address flaws in $C_j$ will never introduce flaws in $C_i$, and so on.

So far, we have not discussed which flaw to address in each flawed state, demanding instead a nonempty set of actions $A(f, \sigma)$ for each flaw $f$ present in a state $\sigma$. We discuss the reason for this in Section 4.3. For now, suffice it to say that we consider algorithms that employ an arbitrary ordering $\pi$ of $F$ and in each flawed state $\sigma$ address the greatest flaw according to $\pi$ in a subset of $U(\sigma)$.
Definition 1. If \( \pi \) is any ordering of \( F \), let \( I_\pi : 2^F \to F \) be the function mapping each subset of \( F \) to its greatest element according to \( \pi \), with \( I_\pi(\emptyset) = \emptyset \). We will sometimes abuse notation and, for a state \( \sigma \in \Omega \), write \( I_\pi(\sigma) \) for \( I_\pi(U(\sigma)) \) and also write \( I \) for \( I_\pi \) when \( \pi \) is clear from context.

Definition 2. Let \( D_\pi \subseteq D \) be the result of retaining for each state \( \sigma \) only the outgoing arcs with label \( I_\pi(\sigma) \).

The next definition reflects that, since actions are selected uniformly, the number of actions available to address a flaw, that is, the breadth of the “repertoire,” is important.

Amenability. The amenability of a flaw \( f \) is

\[
A_f = \min_{\sigma \in f} |A(f, \sigma)|.
\] (4)

The amenability of a flaw \( f \) will be used to bound from below the amount of randomness consumed every time \( f \) is addressed. (The minimum in Equation (4) is often inoperative with \( |A(f, \sigma)| \) being the same for all \( \sigma \in f \).)

3. STATEMENT OF RESULTS

Our first result concerns the simplest case, in which, after choosing a single fixed permutation \( \pi \) of the flaws, in each flawed state \( \sigma \), the algorithm simply addresses the greatest flaw present in \( \sigma \) according to \( \pi \), that is, the algorithm is the uniform random walk on \( D_\pi \).

Theorem 1. If for every flaw \( f \in F \),

\[
\sum_{g \in \Gamma(f)} \frac{1}{A_g} < \frac{1}{e},
\]

then, for any ordering \( \pi \) of \( F \) and any \( \sigma_1 \in \Omega \), the uniform random walk on \( D_\pi \) starting at \( \sigma_1 \) reaches a sink within \((\log_2 |\Omega| + |U(\sigma_1)| + s) / \delta \) steps with probability at least \( 1 - 2^{-s} \), where \( \delta = 1 - \max_{f \in F} \sum_{g \in \Gamma(f)} \frac{A_g}{A_f} \).

Theorem 1 has three features worth discussing, shared by all our further results below.

Arbitrary initial state. The fact that \( \sigma_1 \) can be arbitrary means that any foothold on \( \Omega \) suffices to apply the theorem, without needing to be able to sample from \( \Omega \) according to some measure. While sampling from \( \Omega \) has generally not been an issue in existing applications of the LLL, as we discuss in Section 4, this has only been true precisely because the sets and the measures considered have been highly structured.

Arbitrary number of flaws. The runtime depends only on the number of flaws present in the initial state, \( |U(\sigma_1)| \), not on the total number of flaws \( |F| \). This has an implication analogous to the result of Haeupler et al. [2011] on core events: even when \( |F| \) is very large, for example, superpolynomial in the problem’s encoding length, we can still get an efficient algorithm if we can show that \( |U(\sigma_1)| \) is small, for example, by proving that in every state only polynomially many flaws may be present. This feature provides great flexibility in the design of flaws, as demonstrated in one of our applications, presented in Section 10.

Cutoff phenomenon. The bound on the runtime is sharper than a typical high probability bound, being instead akin to a mixing time cutoff bound [Diaconis 1996], wherein the distance to the stationary distribution drops from near 1 to near 0 in a very small number of steps past a critical point. In our setting, the walk first makes a certain
number of steps without any guarantee of progress; from that point on, every single step has constant probability of being the last step. While, pragmatically, a high probability bound would be just as useful, the fact that our bound naturally takes this form suggests a potential deeper connection with the theory of Markov chains.

Theorem 1 follows from the following significantly more general result. We present the derivation of Theorem 1 from Theorem 2 in Section 6. Observe the similarity between the condition of Theorem 2 and the condition (1) of the general LLL with $1/A_f$ replacing $\Pr(A_i)$.

**Theorem 2 (Main result).** If there exist positive real numbers $\{\mu_f\}$ such that, for every flaw $f \in F$,

$$\frac{1}{A_f} \prod_{g \in \Gamma(f)} (1 + \mu_g) < \mu_f, \quad (5)$$

then, for any ordering $\pi$ of $F$ and any $\sigma_1 \in \Omega$, the uniform random walk on $D_\pi$ starting from $\sigma_1$ reaches a sink within $(T_0 + s)/\delta$ steps with probability at least $1 - 2^{-s}$, where

$$\delta = 1 - \max_{f \in F} \left( \frac{1}{\mu_f A_f} \prod_{g \in \Gamma(f)} (1 + \mu_g) \right) > 0,$$

$$T_0 = \log_2 |\Omega| + |U(\sigma_1)| \cdot \log_2 \left( 1 + \frac{\max_{f \in F}(\mu_f A_f)}{\min_{f \in F}(A_f)} \right).$$

**Remark 2.** In applications, typically, $\delta = \Theta(1)$ and $T_0 = O(\log |\Omega| + |U(\sigma_1)| \log |F|)$.

**Theorem 3.** Under the conditions of Theorem 2, the digraph $D_\pi$ has at least $|\Omega|/\prod_{f \in F}(1 + \mu_f)$ sinks.

### 3.1. Dense Neighborhoods

In a number of applications, the subgraph induced by the neighborhood of each flaw in the causality graph contains several arcs. We improve Theorem 2 in such settings by employing a recursive algorithm. This has the effect that the flaw addressed in each step depends on the entire trajectory up to that point, not just the current state, that is, the walk in non-Markovian. It is for this reason that we required a nonempty set of actions for every flaw present in a state, and why the definition of the causality digraph does not involve flaw choice. Specifically, for any ordering $\pi$ of $F$ and any $\sigma_1 \in \Omega$, the recursive walk is the non-Markovian random walk on $\Omega$ that occurs by invoking procedure `Eliminate` (see Algorithm 1). Observe that, if in Line 8 we did not intersect $U(\sigma)$ with $\Gamma(f)$, the recursion would be trivialized and the walk would be the uniform random walk on $D_\pi$. This is because the first time any “while” condition would be satisfied, causing the corresponding recursive call to return, would be when $U(\sigma) = \emptyset$.

**Definition 3.** Let $G(F, E)$ be the undirected graph on $F$, where $\{f, g\} \in E$ if and only if both $f \to g$ and $g \to f$ exist in the causality digraph $C$. For any $S \subseteq F$, let $\text{Ind}(S) = \{S' \subseteq S : S' \text{ is an independent set in } G\}$.

Observe that, trivially, the condition of Theorem 2 can be restated as requiring that for every flaw $f \in F$,

$$\frac{1}{\mu_f A_f} \sum_{S \subseteq \Gamma(f)} \prod_{g \in S} \mu_g < 1, \quad (6)$$
ALGORITHM 1: Recursive Walk

1: procedure ELIMINATE
2: \( \sigma \leftarrow \sigma_1 \)
3: while \( U(\sigma) \neq \emptyset \) do
4: \( \text{ADDRESS}(I_\pi(\sigma), \sigma) \)
5: return \( \sigma \)

6: procedure ADDRESS\((f, \sigma)\)
7: \( \sigma \leftarrow \text{A uniformly random element of } A(f, \sigma) \)
8: while \( B = U(\sigma) \cap \Gamma(f) \neq \emptyset \) do
9: \( \text{ADDRESS}(I_\pi(B), \sigma) \)

where, throughout, we use the convention that a product devoid of factors equals 1, that is, \( \prod_{x \in \emptyset} f(x) = 1. \)

**Theorem 4.** If there exist positive real numbers \( \{\mu_f\} \) such that, for every flaw \( f \in F, \)

\[
\theta_f := \frac{1}{\mu_f A_f} \sum_{S \in \text{Ind}(\Gamma(f))} \prod_{g \in S} \mu_g < 1, \tag{7}
\]

then, for any ordering \( \pi \) of \( F \) and any \( \sigma_1 \in \Omega, \) the recursive walk on \( D \) starting at \( \sigma_1 \) reaches a sink within \((T_0 + s)/\delta)\) steps with probability at least \( 1 - 2^{-s} \), where \( \delta = 1 - \max_{f \in F} \theta_f, \) and

\[
T_0 = \log_2 |\Omega| + \left( \max_{S \in \text{Ind}(U(\sigma_1))} |S| \right) \cdot \log_2 \left( 1 + \frac{\max_{f \in F} (\mu_f A_f)}{\min_{f \in F} A_f} \right).
\]

**Remark 3.** Theorem 4 improves Theorem 2 in that (i) the summation in Equation (7) is only over the subsets of \( \Gamma(f) \) that are independent in \( G \), instead of being over all subsets of \( \Gamma(f) \) as in Equation (6), and (ii) \( T_0 \) is proportional only to the size of the largest independent subset of \( U(\sigma_1) \) rather than to the size of \( U(\sigma_1) \).

**Remark 4.** Theorem 4 can be strengthened by introducing for each flaw \( f \in F \) a permutation \( \pi_f \) of \( \Gamma(f) \) and replacing \( \pi \) with \( \pi_f \) in Line 9 of Algorithm 1. With this change, in Equation (7), it suffices to sum only over \( S \subseteq \Gamma(f) \), satisfying the following: if the subgraph of \( C \) induced by \( S \) contains an arc \( g \rightarrow h \), then \( \pi_f(g) \geq \pi_f(h) \). As such, a subgraph cannot contain both \( g \rightarrow h \) and \( h \rightarrow g \); thus, we see that \( S \in \text{Ind}(\Gamma(f)) \).

### 3.2. A Left-Handed Algorithm

While Theorems 1 to 4 do not care about the flaw ordering \( \pi \), inspired by the so-called lefthanded version of the LLL introduced by Pegden [2011], we give a condition under which the flaw order \( \pi \) can be chosen in a *provably* beneficial way. This is done by organizing the flaws in an order akin to an elimination sequence. Specifically, the idea is to seek a permutation \( \pi \) and a “responsibility digraph” \( R \), derived from the causality digraph \( C \), to “shift responsibility” from flaws failing to satisfy condition (5) of Theorem 2, to flaws that have slack.

**Definition 4.** For an ordered set of vertices \( v_1 < v_2 < \cdots < v_n \), say that arc \( v_i \rightarrow v_j \) is forward if \( i < j \) and backward if \( i > j \). Given a causality digraph \( C = C(\Omega, F, D) \) and a permutation \( \pi \) of \( F \) ordering the vertices of \( C \), we say that \( R \) is a responsibility digraph for \( C \) with respect to \( \pi \) if:

1. Every forward arc and self-loop of \( C \) exists in \( R \).
2. If a backward arc \( v_j \rightarrow v_i \) of \( C \) does not exist in \( R \), then, for each \( k \) such that \( v_k \rightarrow v_j \) exists in \( R \), \( v_k \rightarrow v_i \) exists in \( R \) as well.
The neighborhood of a flaw $f$ in a responsibility graph $R$ is $\Gamma_R(f) = \{ g \in F : f \rightarrow g \text{ exists in } R \}$.

For any permutation $\pi$ of $F$, any responsibility digraph $R$ with respect to $\pi$, and any $\sigma_1 \in \Omega$, the left-handed walk is the random walk induced on $\Omega$ by modifying the Recursive Walk, as follows.

**ALGORITHM 2: Left-Handed Walk**

In Line 8 of Algorithm 1, replace $\Gamma$ with $\Gamma_R$.

**THEOREM 5.** For any permutation $\pi$ of $F$ and any responsibility digraph $R$ with respect to $\pi$, if there exist positive real numbers $\{\mu_f\}$ such that, for every flaw $f \in F$,

$$\theta_f := \frac{1}{\mu_f A_f} \sum_{S \subseteq \Gamma_R(f)} \prod_{g \in S} \mu_g < 1,$$

then, for any $\sigma_1 \in \Omega$, the left-handed walk on $D$ starting at $\sigma_1$ reaches a sink within $(T_0 + s)/\delta$ steps with probability at least $1 - 2^{-s}$, where $\delta = 1 - \max_{f \in F} \theta_f$, and

$$T_0 = \log_2 |\Omega| + |U(\sigma_1)| \cdot \log_2 \left( 1 + \frac{\max_{f \in F} (\mu_f A_f)}{\min_{f \in F} A_f} \right).$$

**Remark 5.** Since the causality digraph $C$ is, trivially, a responsibility graph, Theorem 5 can be seen as a non-Markovian generalization of Theorem 2, in which flaw choice is driven by recursion and $R$.

**4. COMPARISON TO THE LOVÁSZ LOCAL LEMMA**

In addition to dispensing with the need for $\Omega$ to have product structure (variables) or symmetry (permutations), our setting has two other benefits.

**4.1. Moving Beyond Measure-Preserving Transformations**

The LLL, framed as a result in probability, begins with a probability measure on the set of objects $\Omega$. In terms of proving the existence of flawless objects, its value lies in that it delivers strong results even when the measure is chosen without any consideration of the flaws (bad events). Indeed, most LLL applications simply employ the uniform measure on $\Omega$, a property that can render the LLL indistinguishable from magic. It is worth noting that, in the presence of variables, the uniform measure is nothing but the product measure generated by sampling each variable according to the uniform measure on its domain.

All algorithmic versions of the LLL up to now can be seen as walks on $\Omega$ constrained by the measure. For product measures, that is, in the setting of Moser and Tardos [2010], this means that the only transformation allowed is resampling all variables of a bad event, with each variable resampled independently of all others, using the same distribution every time the variable is resampled obliviously to the current state. The partial resampling framework of Harris and Srinivasan [2013] refines this to allow resampling of a subset of an event’s variables but, again, only independently of one another and obliviously to the current state. Similarly, for the uniform measure on permutations [Harris and Srinivasan 2014]: the permuted elements whose images form a violated constraint must be reshuffled in a very specific and state-oblivious way, mandated by consistency with the uniform measure.

The framework of Harvey and Vondrák [2015] is similarly constrained by measure preservation. Specifically, resampling oracles for a given measure can be seen as action
digraphs that do not have to be atomic and in which the different arcs leaving each vertex (actions) can have different probabilities (weights). On the other hand, the arcs and their weights must satisfy the following very stringent requirement: for every flaw \( f \in F \), if \( \sigma \in f \) is sampled according to the measure and \( f \) is addressed at \( \sigma \), the distribution of the resulting state must be identical to the measure. In other words, each step of the walk should completely remove the conditional induced by the presence of the flaw addressed (while the potential causality digraph should equal the lopsided dependency digraph).

In contrast, our framework dispenses with the measure on \( \Omega \) altogether, allowing the set of transformations for addressing each flaw to depend arbitrarily on the current state (but mandating that actions must be drawn uniformly). This has three distinct effects:

—It allows us to deal with settings in which both the set of objects \( \Omega \) and the set of flaws are amorphous, as in the case of rainbow Hamilton cycles and rainbow perfect matchings, something not possible with any previous algorithmic LLL results.
—In the case of permutations, in which some structure is present, we derive the same main results as Harris and Srinivasan [2014], with dramatically simpler proofs. Moreover, we have far greater freedom in the choice of algorithms since there is no constraint imposed by some measure.
—Finally, for the variable setting of Moser and Tardos [2010], we gain “adaptivity to state.” This allows us to address one of the oldest and most vexing concerns about the LLL (see the survey of Szegedy [2013]), exemplified by the LLL’s inability to establish the elementary fact that a graph with maximum degree \( \Delta \) can be colored with \( q = \Delta + 1 \) colors. Specifically, imagine that, to recolor a monochromatic edge \( e \), we select an endpoint \( v \) of \( e \) arbitrarily and assign a new color \( c \) to \( v \). When the choice of \( c \) must be uniform among all colors, as mandated when using the uniform measure in the variable setting, the obliviousness of the choice necessitates the use of a large number of colors relative to \( \Delta \) in order for new violations to become sufficiently rare for the method to terminate. Specifically, the LLL can only work when \( q > e\Delta \).

On the other hand, in our setting, the color \( c \) can be selected uniformly among the available colors for \( v \), that is, the colors not appearing in \( v \)'s neighborhood, by taking the set of actions \( A(f, \sigma) \) to be precisely the set of states that result by assigning available colors to \( v \) in \( \sigma \). Thus, as soon as \( q \geq \Delta + 1 \), the causality digraph becomes empty and rapid termination follows trivially.

4.2. Dependencies versus Actions

Unlike the variable setting of Moser and Tardos [2010], in which the dependency relation between events is symmetric, our causality relation, similar to the lopsided LLL, is not symmetric. We consider asymmetry a significant structural feature of our work since, as is well known [Szegedy 2013], the directed setting is strictly stronger than the undirected setting. For example, there exist systems of events for which there exists a lopsided dependence digraph sparser than any undirected dependence graph. Moreover, asymmetry is essential in our development of structured clause choice in the left-handed version of our theorem.

At a high level, our results capture the directedness of the lopsided LLL, but with the far more flexible causality digraph replacing the lopsided dependence digraph. Concretely, our framework replaces the limited negative dependence condition of the lopsided LLL, which can be highly nontrivial to establish [Lu et al. 2013], with limited causality under atomicity, a condition that is both significantly less restrictive and far easier to check. Moreover, as mentioned earlier and to our pleasant surprise, in...
all applications that we have considered so far, we have found atomicity to be a very valuable aid in the design of flaws and actions.

For example, in Section 8, we give the first efficient algorithm for finding rainbow Hamilton cycles in hypergraphs, as guaranteed to exist by the nonconstructive results of Dudek et al. [2012] and Dudek and Ferrara [2015]. When we tried to determine flaws and actions for this setting, to our delight, we realized that we could just use one of the main technical propositions of Dudek and Ferrara [2015], as it is equivalent to proving that for each flaw \( f \) and \( \sigma \in f \) there exists a set of actions such that the corresponding digraph \( D \) is atomic. As Dudek and Ferrara [2015] is completely independent of our work, we consider this “coincidence” a nice testament to the naturalness of atomicity.

In a different direction, in Section 10, we give an application regarding the Color-Blind index of Graphs. That setting highlights the importance of the directedness of the causality graph and demonstrates how directedness readily enables the formulation of “obvious” flaws and actions. Finally, our Theorem 4 combines the benefits of directedness with the improvement of Bissacot et al. [2011], by restricting the summation to independent sets. For example, in Section 11, we show how Theorem 4 allows us to also give an efficient algorithm for Latin Transversals matching the \( \Delta_1 \leq (27/256)n \) bound of Harris and Srinivasan [2014]. (Theorem 4 can also benefit the application to rainbow matchings in Section 9.1, but we chose to use Theorem 1 to keep the exposition simple.)

While the main contribution of our framework lies in providing freedom in the design of the set of actions for addressing each flaw in each state (thus going beyond the LLL), its main limitation is that we are restricted in performing uniform random walks in the corresponding directed graph \( D \). That means, for example, that our framework does not capture applications of the variable setting in which the product measure is not uniform over the domain of each variable, while the variable setting of Moser and Tardos [2010] captures these cases. We leave closing this gap to future work.

4.3. Flaw Selection

As mentioned earlier, in Theorems 1 to 4, the necessary condition is independent of the flaw order \( \pi \); therefore, if the condition is met, the algorithm reaches a sink quickly for every permutation \( \pi \). As this is an unnecessarily luxurious conclusion, it is natural to try to sharpen the results by selecting the flaw order \( \pi \) first, so that the causality digraph is the image of the (much) sparser \( D_\pi \) instead of \( D \). However, since an arc \( f \to g \) will exist in the causality digraph \( C \) as long as there is even one transition addressing \( f \) that causes \( g \) in \( D_\pi \), it is not at all clear that sparsifying \( D \) using a generic \( \pi \) helps significantly. At the same time, if there exists a “special” \( \pi \) that does help significantly, coming up with it is nontrivial. For example, in the setting of satisfiability, if \( f, g \) are clauses that share variable \( v \) with opposite signs, then not having the arc \( f \to g \) in \( C \) requires either that addressing \( f \) should never involve flipping \( v \), cutting \( A_f \) by half, or finding a permutation of the clauses such that in every state in which \( f \) is the greatest violated clause, \( g \) is satisfied by some variable other than \( v \). The only nontrivial case we know in which the latter can be done is when \( F \) is satisfiable by the pure literal heuristic.

As far as we know, the method by which a bad event (flaw) is selected in each step does not affect the performance of any of the algorithmic extensions of the LLL even though, in the setting of Moser and Tardos [2010], this choice can be arbitrary. The only use we know of this freedom lies in enabling parallelization when \( \Omega \) is a structured set, that is, when \( \Omega \) has product structure [Moser and Tardos 2010; Kolipaka and Szegedy 2011; Chandrasekaran et al. 2013], or it is a set of permutations [Harris and Srinivasan 2014]. Since we allow \( \Omega \) to be completely amorphous, it is not readily clear how to approach parallelization in our setting.
Finally, we note that flaw choice in our framework is not really restricted to using a single permutation. For example, in the non recursive setting, before beginning the walk, we can select an arbitrary infinite sequence of permutations $\pi_1, \pi_2, \ldots$ of $F$ and in the $i$th step of the walk address the greatest flaw present according to $\pi_i$. If $\pi_1 = \pi_2 = \cdots$ we are back to the single-permutation setting, while if, for example, each $\pi_i$ is an independent uniformly random permutation, the algorithm addresses a uniformly random flaw present in each step. At the same time, we must make clear that our framework does not accommodate arbitrary flaw selection functions and, in fact, we do not see how to extend it beyond permutation-based choice. To keep the presentation of our results uniform (and compact), we have stated both Theorems 2 and 5 in terms of a single permutation. We do point out the one place in our proofs that changes (trivially) to handle multiple permutations.

5. MAPPING BAD TRAJECTORIES TO FORESTS

We prove Theorems 2 to 5 in three parts. In the first part, carried out in this section, we show how to represent each sequence of $t$ steps that does not reach a sink as a forest with $t$ vertices, in which the forests have different characteristics for each of the walks of Theorems 2 to 5. Then, in Section 6, we state a general lemma for bounding the runtime of different walks in terms of properties of their corresponding forests and show how it readily implies each of Theorems 2 to 5. Finally, in Section 7, we prove the lemma itself. In a first reading, the reader may want to skip Section 5.3 (and, perhaps, also Section 5.2). The sections can be read later, in order, after the material of Section 5.1 has been absorbed.

In the following, to lighten notation, we will assume that $\sigma_1 \in \Omega_1$ is fixed but arbitrary.

Definition 5. A walk $\Sigma = \sigma_1 \xrightarrow{w_1} \sigma_2 \xrightarrow{w_2} \sigma_3 \cdots \sigma_t \xrightarrow{w_t} \sigma_{t+1}$ is called a $t$-trajectory. A $t$-trajectory is bad if it only goes through flawed states. Let $\text{Bad}(t)$ be the set of bad $t$-trajectories starting at $\sigma_1$. 

Our first step is the same as Moser’s [Moser 2009], generalized to the notion of atom- icity. It amounts to defining an almost-1-to-1 map from bad $t$-trajectories to sequences of $t$ flaws. While the map is not 1-1, crucially, it becomes 1-1 with the addition of a piece of information whose size is independent of $t$.

Definition 6. If $\Sigma = \sigma_1 \xrightarrow{w_1} \sigma_2 \xrightarrow{w_2} \sigma_3 \cdots \sigma_t \xrightarrow{w_t} \sigma_{t+1}$ is a bad $t$-trajectory, the sequence $W(\Sigma) = w_1, w_2, \ldots, w_t$, that is, the sequence of flaws labeling the arcs $\Sigma$, is the witness of $\Sigma$.

Claim 1. If $D$ is atomic, then the map from bad $t$-trajectories $\Sigma \rightarrow \langle W(\Sigma), \sigma_{t+1}\rangle$ is one to one.

Proof. The atomicity of $D$ implies that $\sigma_t$ is the unique state in $\Omega$ with an arc $\sigma_t \xrightarrow{w_t} \sigma_{t+1}$, and so on. □

Thus, $|\text{Bad}(t)|$ is bounded by the number of possible witness $t$-sequences multiplied by $|\Omega|$. 

5.1. Forests of the Uniform Walk (Theorem 2)

Recall that, for any ordering $\pi$ of the flaws, we denote by $D_\pi$ the digraph that results from $D$ if at each state $\sigma$ we only retain the outgoing arcs labeled by $I_\pi(\sigma) = I(\sigma)$. To analyze the uniform random walk on $D_\pi$ we will represent witnesses as sequences of sets reflecting causality.
Let $B_i$ be the set of flaws “introduced” by the $i$th step of the walk, where a flaw $f$ is said to “introduce itself” if it remains present after an action from $A(f, \sigma_i)$ is taken. Formally,

**Definition 7.** Let $B_0 = U(\sigma_1)$. For $1 \leq i \leq t - 1$, let $B_i = U(\sigma_{i+1}) \setminus (U(\sigma_i) \setminus I(\sigma_i))$.

Let $B_i^* \subseteq B_i$ comprise those flaws addressed in the course of the trajectory. Thus, $B_i^* = B_i \setminus (O_i \cup N_i)$, where $O_i$ comprises any flaws in $B_i$ that were eradicated “collaterally” by an action taken to address some other flaw, and $N_i$ comprises any flaws in $B_i$ that remained present in every subsequent state after their introduction without being addressed. Formally,

**Definition 8.** The Break Sequence of a bad $t$-trajectory is $B_0^*, B_1^*, \ldots, B_{t-1}^*$, where, for $0 \leq i \leq t - 1$,

$$
O_i = \{ f \in B_i \mid \exists j \in [i + 1, t) : f \notin U(\sigma_{j+1}) \land \forall \ell \in [i + 1, j) : f \neq w_\ell \},
$$

$$
N_i = \{ f \in B_i \mid \forall j \in [i + 1, t) : f \notin U(\sigma_{j+1}) \land \forall \ell \in [i + 1, t) : f \neq w_\ell \},
$$

$$
B_i^* = B_i \setminus (O_i \cup N_i).
$$

Given $B_0^*, B_1^*, \ldots, B_{t-1}^*$, we can determine $w_1, w_2, \ldots, w_i$ inductively, as follows. Define $E_1 = B_0^*$, while, for $i \geq 1$,

$$
E_{i+1} = (E_i - w_i) \cup B_i^*.
$$

By construction, the set $E_i \subseteq U(\sigma_i)$ is guaranteed to contain $w_i = I(\sigma_i) = I(U(\sigma_i))$. Since $I = I_\pi$ returns\(^1\) the greatest flaw in its input according to $\pi$, it must be that $I_\pi(E_i) = w_i$. We note that this is the only place we ever make use of the fact that the function $I$ is derived by an ordering of the flaws, thus guaranteeing, that for every $f \in F$ and $S \subseteq F$, if $I(S) \neq f$, then $I(S \setminus f) = I(S)$.

We next give another 1-to-1 map, mapping each Break Sequence to a vertex-labeled rooted forest. Specifically, the Break Forest of a bad $t$-trajectory $\Sigma$ has $|B_0^*|$ trees and $t$ vertices, each vertex labeled by an element of $W(\Sigma)$. To construct it, we first lay down $|B_0^*|$ vertices as roots and then process the sets $B_1^*, B_2^*, \ldots$ in order, each set becoming the progeny of an already existing vertex (empty sets, thus giving rise to leaves).

**ALGORITHM 3:** Break Forest Construction

1. Lay down $|B_0^*|$ vertices, each labeled by a different element of $B_0^*$, and let $V$ consist of these vertices
2. for $i = 1$ to $t - 1$
3. Let $v_i$ be the vertex in $V_i$ with greatest label according to $\pi$
4. Add $|B_i^*|$ children to $v_i$, each labeled by a different element of $B_i^*$
5. Remove $v_i$ from $V$; add to $V$ the children of $v_i$

Observe that, even though neither the trees nor the nodes inside each tree of the Break Forest are ordered, we can still reconstruct $W(\Sigma)$ since the set of labels of the vertices in $V_i$ equals $E_i$ for all $0 \leq i \leq t - 1$.

### 5.2. Forests of the Recursive Walk (Theorem 4)

We will represent each bad $t$-trajectory, $\Sigma$, of the Recursive Walk (Algorithm 1) as a vertex-labeled unordered rooted forest, having one tree per invocation of procedure ADDRESS by procedure ELIMINATE. Specifically, to construct the Recursive Forest $\phi = \phi(\Sigma)$,

---

\(^1\)If, instead of $\pi$, we had a sequence of permutations $\pi_1, \pi_2, \ldots$, we would simply use $I_{\pi_i}$ to determine $w_i$ from $E_i$. 

---

Journal of the ACM, Vol. 63, No. 3, Article 22, Publication date: July 2016.
we add a root vertex per invocation of \texttt{address} by \texttt{eliminate} and one child to every vertex for each (recursive) invocation of \texttt{address} that it makes. As each vertex corresponds to an invocation of \texttt{address} (step of the walk), it is labeled by the invocation’s flaw argument. Observe now that (the invocations of \texttt{address} corresponding to) both the roots of the trees and the children of each vertex appear in \Sigma in their order according to \pi. Thus, given the unordered rooted forest \phi(\Sigma), we can order its trees and the progeny of each vertex according to \pi and recover \(W(\Sigma)\) as the sequence of vertex labels in the preorder traversal of the resulting ordered rooted forest.

Recall the definition of graph \(G\) on \(F\) from Definition 3. We will prove that the flaws labeling the roots of a Recursive Forest are independent in \(G\) and that the same is true for the flaws labeling the progeny of every vertex of the forest. To do this, we first prove the following.

**Proposition 1.** If \(\text{address}(f, \sigma)\) returns at state \(\tau\), then \(U(\tau) \subseteq U(\sigma) \setminus (\Gamma(f) \cup \{f\})\).

**Proof.** Let \(\sigma'\) be any state subsequent to the \(\text{address}(f, \sigma)\) invocation. If any flaw in \(U(\sigma) \cap \Gamma(f)\) is present at \(\sigma'\), the “while” condition in Line 8 of the Recursive Walk (Algorithm 1) prevents \(\text{address}(f, \sigma)\) from returning. On the other hand, if \(h \in \Gamma(f) \setminus U(\sigma)\) is present in \(\sigma'\), then there must have existed an invocation \(\text{address}(g, \sigma'')\), subsequent to invocation \(\text{address}(f, \sigma)\), wherein addressing \(g\) caused \(h\). Consider the last such invocation. If \(\sigma''\) is the state when this invocation returns, then \(h \not\in U(\sigma'')\), for otherwise the invocation could not have returned, and by the choice of invocation, \(h\) is not present in any subsequent state between \(\sigma''\) and \(\tau\). \(\Box\)

Let \((f_i, \sigma_i)\) denote the argument of the \(i\)th invocation of \texttt{address} by \texttt{eliminate}. By Proposition 1, \(\{U(\sigma_i)\}_{i \geq 1}\) is a decreasing sequence of sets. Thus, the claim regarding the root labels follows trivially: for each \(i \geq 1\), the flaws in \(\Gamma(f_i) \cup f_i\) are not present in \(\sigma_{i+1}\), therefore are not present in \(U(\sigma_j)\) for any \(j \geq i + 1\). The proof for the children of each node is essentially identical. If a node corresponding to an invocation \(\text{address}(f, \sigma)\) has children corresponding to (recursive) invocations with arguments \(\{g_i, \sigma_i\}\), then the sequence of sets \(\{U(\sigma_i)\}_{i \geq 1}\) is decreasing. Thus, the flaws in \(\Gamma(g_i) \cup g_i\) are not present in \(\sigma_{i+1}\), therefore are not present in \(U(\sigma_j)\) for any \(j \geq i + 1\).

### 5.3. Forests of the Left-Handed Walk (Theorem 5)

Recall that \(\pi\) is an arbitrary permutation of \(F\) and that the Left-Handed Walk (Algorithm 2) is the Recursive Walk (Algorithm 1) modified by replacing \(\Gamma(f)\) with \(\Gamma_R(f)\) in Line 8, where \(R\) is a responsibility graph for \(D\) with respect to \(\pi\). We map the bad trajectories of the Left-Handed Walk into vertex-labeled unordered rooted forests, exactly as we did for the bad trajectories of the Recursive Walk, that is, one tree per invocation of \texttt{address} by \texttt{eliminate}, one child per recursive invocation of \texttt{address}, all vertices labeled by the flaw argument of the invocation. The challenge for the Left-Handed Walk is to prove that the labels of the roots are distinct and, similarly, that the labels of the children of each node are distinct. (For Break Forests (Algorithm 3), both properties were true automatically; for Recursive Forests, we established the stronger property that each of these sets of flaws is independent.) To do this, we first prove the following analogue of Proposition 1.

**Definition 9.** Let \(S_f\) denote the set of flaws strictly greater than \(f\) according to \(\pi\). For a state \(\sigma\) and a flaw \(f \in U(\sigma)\), let \(W(\sigma, f) = U(\sigma) \cap S_f\).

**Proposition 2.** If \(\text{address}(f, \sigma)\) returns at state \(\tau\), then \(\tau \not\in f\) and \(W(\tau, f) \subseteq W(\sigma, f)\).

**Proof.** The execution of \(\text{address}(f, \sigma)\) generates a recursion tree, each node labeled by its flaw argument. Thus, the root is labeled by \(f\) and each child of the root is labeled.
by a flaw in $\Gamma_R(f)$. Let $S_f^+ = S_f \cup \{f\}$. For a state $\omega$, let $Q(f, \omega)$ be the set of flaws in $S_f^+ \setminus \Gamma_R(f)$ that are present in $\omega$. We claim that, if $g \in \Gamma_R(f)$ and $\text{ADDRESS}(g, \omega)$ terminates at $\omega'$, then $Q(f, \omega') \subseteq Q(f, \omega)$. This suffices to prove the lemma since:

—By the claim, any flaw in $Q(f, \tau) \setminus Q(f, \sigma)$ must be introduced by the action $\sigma \xrightarrow{f} \sigma'$ taken by the original invocation $\text{ADDRESS}(f, \sigma)$. Thus, $Q(f, \tau) \subseteq Q(f, \sigma')$.

—All flaws in $S_f^+$ introduced by $\sigma \xrightarrow{f} \sigma'$ are in $\Gamma_R(f)$, since $R$ contains all forward edges and self-loops of $C$. Thus, $Q(f, \sigma') \subseteq Q(f, \sigma)$. In particular, $f$ can only be present in $\sigma'$ if $f \in \Gamma_R(f)$.

—No flaw in $\Gamma_R(f)$ can be present in $\tau$ since $\text{ADDRESS}(f, \sigma)$ returned at $\tau$.

To prove the claim, consider the recursion tree of $\text{ADDRESS}(g, \omega)$. If $h \in Q(f, \omega')$ and $h \notin Q(f, \omega)$, then there has to be a path $g_1 = g, g_2, \ldots, g_i$ from the root of the recursion tree of $\text{ADDRESS}(g, \omega)$ to a node $g_i$ such that $h \in \Gamma(g_i)$ but $h \notin \Gamma_R(g_j)$ for each $j \in [i]$. To see this, note that, since $h$ was absent in $\omega$ but is present in $\omega'$, it must have been introduced by some flaw $g_i$ addressed during the execution of $\text{ADDRESS}(g, \omega)$. But if $h$ belonged in the neighborhood with respect to $R$ of any of the flaws on the path from the root to $g_i$, the algorithm would have not terminated. However, such a path cannot exist, as it would require all of the following to be true, violating the definition of responsibility digraphs (let $g_0 = f$ for notational convenience): (i) $h \in \Gamma(g_i)$, (ii) $h \notin \Gamma_R(g_i)$, (iii) $g_i \in \Gamma_R(g_{i-1})$, and (iv) $h \notin \Gamma_R(g_{i-1})$. □

To establish the distinctness of the root labels, observe that each time procedure $\text{ELIMINATE}$ is invoked at a state $\sigma$, by definition of $I_\pi$, we have $W(\sigma, (I_\pi(\sigma))) = \emptyset$. By Proposition 2, if the invocation returns at state $\tau$, then neither $I_\pi(\sigma)$ nor any greater flaws are present in $\tau$. Therefore, $\text{ELIMINATE}$ invokes $\text{ADDRESS}$ at most once for each $f \in F$. To see the distinctness of the labels of the children of each node, consider an invocation of $\text{ADDRESS}(f, \sigma)$. Whenever this invocation recursively invokes $\text{ADDRESS}(g, \sigma')$, where $g \in \Gamma_R(f)$, by definition of $I_\pi$, every flaw in $S_g \cap \Gamma_R(f)$ is absent from $\sigma'$. By Proposition 2, whenever each such invocation returns, neither $g$ nor any of the flaws in $S_g \cap \Gamma_R(f)$ are present, implying that $\text{ADDRESS}(f, \sigma)$ invokes $\text{ADDRESS}(g, \sigma')$ at most once for each $g \in \Gamma_R(f)$.

6. A GENERAL FOREST LEMMA AND PROOF OF THEOREMS 1 TO 5

Recall that we are considering random walks on the multidigraph $D$ on $\Omega$, which has an arc $\sigma \xrightarrow{f} \tau$ for each $\sigma \in \Omega$, flaw $f \ni \sigma$, and $\tau \in A(f, \sigma)$. Recall also that the different walks of Theorems 2 to 5 differ only on which flaw to address among those present in the current state $\sigma$. Having chosen to address a flaw $f \ni \sigma$, all three walks proceed in the exact same manner, selecting the next state $\tau \in A(f, \sigma)$ uniformly at random. In Section 5, we saw how to map the bad trajectories of the different walks into unordered rooted forests so that, given a trajectory’s forest and final state, we can reconstruct it.

Next, we will formulate and prove a general tool for bounding the runtime of different walks on $D$.

**Lemma 1 (Witness Forests).** Consider any random walk on $D$ that (i) in every flawed state $\sigma$, after choosing (arbitrarily) which flaw $f \ni \sigma$ to address, selects the next state $\tau \in A(f, \sigma)$ uniformly at random; and (ii) whose bad trajectories can be mapped into unordered rooted forests satisfying the following properties, so that, given a trajectory’s forest, we can reconstruct the sequence of flaws addressed along the trajectory:

1. Each vertex of the forest is labeled by a flaw $f \in F$.  

Journal of the ACM, Vol. 63, No. 3, Article 22, Publication date: July 2016.
(2) The flaws labeling the roots of the forest are distinct and, as a set, belong in the set \( \text{Roots}(\sigma_1) \subseteq 2^F \).

(3) The flaws labeling the children of each vertex are distinct.

(4) If a vertex is labeled by flaw \( f \), the labels of its children, as a set, belong in the set \( \text{List}(f) \subseteq 2^F \).

If there exist positive real numbers \( \{\mu_f\} \) such that, for every flaw \( f \in F \),

\[
\theta_f := \frac{1}{\mu_f A_f} \sum_{S \in \text{List}(f)} \prod_{g \in S} \mu_g < 1,
\]

then for any \( \sigma_1 \in \Omega \), a walk started at \( \sigma_1 \) reaches a sink within \( (T_0 + s)/\delta \) steps with probability at least \( 1 - 2^{-s} \), where \( \delta = 1 - \max_{f \in F} \theta_f \), and

\[
T_0 = \log_2 |\Omega| + \left( \max_{S \in \text{Roots}(\sigma_1)} |S| \right) \cdot \log_2 \left( 1 + \frac{\max_{f \in F}(\mu_f A_f)}{\min_{f \in F} A_f} \right).
\]

### 6.1. Proof of Theorems 2 to 5 from Lemma 1

Theorem 2 follows immediately by observing that Break Forests trivially satisfy the conditions of Lemma 1 with \( \text{Roots}(\sigma_1) = 2^{U(\sigma_1)} \) and \( \text{List}(f) = 2^{\Gamma(f)} \). Theorem 4 follows by observing that Recursive Forests satisfy the conditions with \( \text{Roots}(\sigma_1) = \text{Ind}(U(\sigma_1)) \) and \( \text{List}(f) = \text{Ind}(\Gamma(f)) \). Theorem 5 follows by observing that Left-Handed Forests satisfy the conditions with \( \text{Roots}(\sigma_1) = 2^{U(\sigma_1)} \) and \( \text{List}(f) = 2^{\Gamma_R(f)} \).

### 6.2. Proof of Theorem 1 from Theorem 2

Let \( Z \geq 1 \) be the least common multiple of the integers \( \{A_f : f \in F\} \). Let \( d := \max_{f \in F} \sum_{g \in \Gamma(f)} \frac{Z}{A_g} \). Observe that \( d \geq \max_{f \in F} |\Gamma(f)| \) since \( |\Gamma(f)| \geq 1 \) for every \( f \in F \), and that for any set \( S \subseteq \Gamma(f) \),

\[
\prod_{g \in S} \frac{Z}{A_g} = \prod_{g \in S} \left( \frac{Z/A_g}{1} \right) \leq \left( \sum_{g \in \Gamma(f)} \frac{Z/A_g}{|S|} \right) \leq \left( \frac{d}{|S|} \right).
\]  

(9)

Taking \( \mu_f = \frac{Z}{(d A_f)} > 0 \) and invoking Equation (9), we see that the hypothesis of Theorem 1 implies that

\[
\frac{1}{\mu_f A_f} \prod_{g \in \Gamma(f)} (1 + \mu_g) = \frac{d}{Z} \sum_{S \subseteq \Gamma(f)} \prod_{g \in S} \frac{Z}{d A_g} = \frac{1 - \delta}{e} \sum_{i=0}^{d} \left( \frac{1}{d} \right)^i \frac{1 - \delta}{e} \left( 1 + \frac{1}{d} \right)^d \leq 1 - \delta.
\]

Regarding the runtime, observe that \( \max_{f \in F}(\mu_f A_f) / \min_{f \in F} A_f \leq (Z/d) / 1 \) since \( |\Gamma(f)| \geq 1 \).

### 6.3. Proof of Theorem 3

The proof of Theorem 3 is quite similar to the inductive proof of the lopsided local lemma. To simplify notation, let \( F = \{f_1, \ldots, f_m\} \) and write \( x_i = \mu_{f_i} / (1 + \mu_{f_i}) \). For any \( S \subseteq [m] \) and \( i \notin S \), let

\[
\text{good}(S) = \bigcap_{j \in S} \overline{f}_j, \quad \text{bad}(i, S) = f_i \cap \text{good}(S), \quad \text{image}(i, S) = \bigcup_{\sigma \in \text{bad}(i, S)} A(i, \sigma).
\]

We will prove that \( |\text{bad}(i, S)| \leq x_i |\text{good}(S)| \) for every \( S \subseteq [m] \) and \( i \notin S \). As this is equivalent to \( |\text{good}(S \cup \{i\})| \geq (1 - x_i) |\text{good}(S)| \) and since \( \text{good}([m]) \) is the set of sinks,
we can conclude that
\[
\frac{|\text{good}(m)|}{|\Omega|} = \frac{|\text{good}(1)|}{|\text{good}(\emptyset)|} \cdot \frac{|\text{good}(2)|}{|\text{good}(1)|} \cdot \frac{|\text{good}(3)|}{|\text{good}(2)|} \cdots \frac{|\text{good}(m)|}{|\text{good}(m-1)|} \geq \prod_{f \in F} (1 - x_i).
\]

Since \( D \) is atomic, \( |\text{image}(i, \cdot)| \geq |\text{bad}(i, \cdot)| \cdot A_f \). Now, let \( B = S \cap \Gamma(i) \) and \( G = S \setminus B \). By the definition of \( G \) and the fact \( i \not\in S \), we see that \( \text{image}(i, G) \subseteq \text{good}(G) \). Therefore,
\[
|\text{bad}(i, S)| \leq |\text{bad}(i, G)| \leq |\text{image}(i, G)| x_i \prod_{j \in \Gamma(i)} (1 - x_j) \leq |\text{good}(G)| x_i \prod_{j \in \Gamma(i)} (1 - x_j). \tag{10}
\]

We can now proceed by induction on \(|S|\). If \( S = \emptyset \), substituting \( G = S \) into Equation (10) concludes the proof. Ditto if \( S \not= \emptyset \) but \( B = \emptyset \). If \( B = \{d_1, \ldots, d_r\} \not= \emptyset \), then, since \(|G| < |S|\), the induction hypothesis implies that
\[
\frac{\text{Good}(S)}{\text{Good}(G)} = \frac{\text{Good}(G \cup \{d_1\})}{\text{Good}(G)} \cdot \frac{\text{Good}(G \cup \{d_1, d_2\})}{\text{Good}(G \cup \{d_1\})} \cdots \frac{\text{Good}(G \cup \{d_1, \ldots, d_r\})}{\text{Good}(G \cup \{d_1, \ldots, d_{r-1}\})} \geq \prod_{j \in B} (1 - x_j). \tag{11}
\]

Since \( B \subseteq \Gamma(i) \), substituting Equation (11) into Equation (10) yields \(|\text{bad}(i, S)| \leq x_i |\text{good}(S)|\).

7. PROOF OF LEMMA 1

7.1. Versions of Flaws

Recall that we are considering random walks on the multidigraph \( D \) on \( \Omega \) which has an arc \( \sigma \xrightarrow{f} \tau \) for each \( \sigma \in \Omega \), flaw \( f \ni \sigma \), and \( \tau \in A(f, \sigma) \). For the proof, it will be convenient to transform \( D \) to another multidigraph \( D^* \), as described below. The transformation is trivial from an algorithmic point of view, but helps with the eventual counting. Let \( Z \geq 1 \) be the least common multiple of the integers \( \{A_f : f \in F\} \).

To form the multidigraph \( D^* \), we replace each arc \( \sigma \xrightarrow{f} \tau \) in \( D \) with \( Z/\text{A}_f \) arcs from \( \sigma \) to \( \tau \), carrying labels \( f_1, f_2, \ldots, f_{Z/\text{A}_f} \). We refer to each such label as a version of flaw \( f \). To move in \( D^* \) exactly as in \( D \), the walk first determines which flaw \( f \ni \sigma \) to address, then chooses \( \tau \in A(f, \sigma) \) uniformly at random. The only difference is that, having done so, now the walk also consumes an additional amount of randomness to “choose a version” of \( f \), that is, to chose one of the \( Z/\text{A}_f \) arcs from \( \sigma \) to \( \tau \). Thus, the probability distribution on sequences of states of the walk in \( D^* \) is identical to the one in \( D \) (indeed, the two walks can be coupled so that the sequences are always equal).

Definition 10. A trajectory \( \Sigma = \sigma_1 \xrightarrow{w_1} \sigma_2 \xrightarrow{w_2} \sigma_3 \cdots \sigma_i \xrightarrow{w_i} \sigma_{i+1} \) on \( D^* \), where \( w_i \) is a version of the flaw addressed at the \( i \)-step, is called a versioned \( t \)-trajectory. A versioned \( t \)-trajectory is bad if it only goes only through flawed states. Let \( \text{VerBad}(t) \) be the set of all versioned bad \( t \)-trajectories.

Observe that, to move in \( D^* \) from any flawed state \( \sigma \) to the next state, the walk must select among
\[
|A(f, \sigma)| \cdot \frac{Z}{\text{A}_f} \geq Z \tag{12}
\]
possibilities, implying that every versioned bad \( t \)-trajectory has probability at most \( 1/Z^t \). Having a uniform upper bound of probability as a function of length is precisely why we introduced versioned flaws.

To prove Lemma 1, we will give \( T_0 = T_0(|\Omega|, U(\sigma_1), \{A_f, \mu_f : f \in F\}) \) such that the probability that a versioned \((T_0 + s)\)-trajectory on \( D^* \) is bad is exponentially small in
s. Per our earlier discussion, to prove this, it suffices to prove that \(|\text{VerBad}(t)|/Z|^{t}\) is exponentially small in \(s\) for \(t = T_0 + s\). Since \(D\) is atomic, we can reconstruct any bad versioned \(t\)-trajectory from \(\sigma_{t-1}\) and the sequence of versioned flaws addressed. We are thus left to count the number of possible sequences of versioned flaws.

Per the hypothesis of Lemma 1, each bad \(t\)-trajectory on \(D\) is associated with a rooted labeled witness forest with \(t\) vertices such that, given the forest, we can reconstruct the sequence of flaws addressed along the \(t\)-trajectory. To count sequences of versioned flaws, we relabel the vertices of the witness forest to carry not only the flaw addressed, but also the integer denoting its version (in the corresponding walk on \(D^*\)). We refer to the resulting object as the versioned witness forest. Recall that neither the trees nor the nodes inside each tree in the witness forest are ordered. To facilitate counting we fix an arbitrary ordering \(\psi\) of \(F\) and map each versioned witness forest into the unique ordered forest that result by ordering the trees in the forest according to the labels of their roots and similarly ordering the progeny of each vertex according to \(\psi\) (recall that both the flaws labeling the roots and the flaws labeling the children of each vertex are distinct).

Having induced this ordering for the purpose of counting, we will encode each versioned witness forest as a rooted, ordered \(d\)-ary forest \(T\) with exactly \(t\) nodes, where \(d = \max_{f \in F} \sum_{g \in \text{List}(f)} Z/A_f\) (recall that \(Z\) is the least common multiple of the integers \(\{A_f : f \in F\}\)). In a rooted, ordered \(d\)-ary forest both the roots and the at most \(d\) children of each vertex are ordered. We think of the root of \(T\) as having reserved for each flaw \(f \in \text{Roots}(\sigma_1)\) a group of \(Z/A_f\) slots, where the \(i\)th group of slots corresponds to the \(i\)th largest flaw in \(F\) according to \(\psi\). If \(f \in \text{Roots}(\sigma_1)\) is the \(i\)th largest flaw in \(F\) according to \(\psi\) and its version in \(V\) is \(j\), then we fill the \(j\)th slot of the \(i\)th group of slots (recall that the flaws labeling the roots of the witness forest are distinct and that, as a set, belong in the set \(\text{Roots}(\sigma_1)\)).

Each node \(v\) of \(T\) corresponds to a node of the witness forest and, therefore, to a flaw \(f\) that was addressed at some point in the \(t\)-trajectory of the algorithm. Recall now that each node in the witness forest that is labeled by a flaw \(f\) has children labeled by distinct flaws in \(\text{List}(f)\). We thus think of each node \(v\) of \(T\) as having precisely \(Z/A_{\sigma_1}\) slots reserved for each flaw \(g \in \text{List}(f)\) (thus, at most, \(d\) reserved slots in total). For each \(g \in \text{List}(f)\) whose version is \(j\), we fill the \(j\)th slot reserved for \(g\) and make it a child of \(v\) in \(T\). Thus, from \(T\), we can reconstruct the sequence of versioned flaws addressed with the algorithm.

At this point, we could proceed and bound \(|\text{VerBad}(t)|\) by the number of all \(d\)-ary ordered forests. Indeed, doing so would yield Theorem 2. Such a counting, though, would ignore the fact that the set of flaws labeling the progeny of a node labeled by \(f\) is not an arbitrary subset of \(F\) but an element of \(\text{List}(f)\). Thus, not every ordered \(d\)-ary forest is a possible versioned witness forest. To quantify this observation, we use ideas from Pegden [2014]. Specifically, we introduce a branching process that produces only ordered \(d\)-ary forests that correspond to versioned witness forests and bound \(|\text{VerBad}(t)|\) by analyzing it. Before describing the branching process, we introduce some conventions and definitions regarding versions of flaws:

—For each \(S \subseteq F\), we will denote by \(\text{Ver}(S)\) the set formed by replacing each \(f \in S\) by its versions \(f_1, f_2, \ldots, f_{Z/A_f}\). For example, \(\text{Ver}(F)\) contains every version of every flaw.

—For each flaw \(f\), we define \(\text{List}'(f)\) to be the set that results by replacing each \(\{g^1, g^2, \ldots, g^k\} \in \text{List}(f)\) by the \(\prod_{i=1}^k Z/A_{g_i}\) sets of the form \(\{g_{i_1}^1, g_{i_2}^2, \ldots, g_{i_k}^k\}\), where \(g_{i_j}^i\) is the \(i\)th version of flaw \(g^i\).

—We assign to each flaw \(f \in F\) a real number \(x_f > 1\)
—Versioned flaws inherit all the features of the underlying flaw. That is, for each $f_i \in \text{Ver}(F)$:
- $x_{f_i} := x_f$
- $\text{List}(f_i) := \text{List}(f)$
- $\text{List}'(f_i) := \text{List}'(f)$

Write $\text{Roots}(\sigma_1) = \text{Roots}$ to simplify notation and let $m = \max_{S \in \text{Roots}} |S|$. Our branching process takes as input an integer $r \leq m$. To start the process we choose an $r$-subset $R$ of $F$ uniformly at random and create $r$ roots, each labeled by a different element of $R$. In each subsequent round, each node $u$ with label $\ell$ “gives birth” by rejection sampling. Specifically, for each versioned flaw $g_i \in \text{Ver}(\text{List}(\ell))$ independently, with probability $1/x_{g_i}$ we add a vertex with label $g_i$ a sibling of $u$. If the resulting set of children of $u$ is in $\text{List}'(\ell)$, we accept the birth. If not, we delete the children created and try again. Note that while the roots of the resulting trees are labeled by flaws, all other nodes are labeled by versioned flaws. It is not hard to see that this process creates every possible versioned witness forest with $r$ unversioned roots with positive probability. Specifically, for a vertex labeled by $\ell$, every set $S \not\in \text{List}'(\ell)$ receives probability 0, while every set $S \in \text{List}'(\ell)$ receives probability proportional to $w_{\ell}(S) = \prod_{g \in S} \frac{1}{x_g} \prod_{h \in \text{Ver}(\text{List}(\ell)) \setminus S} \left(1 - \frac{1}{x_h}\right)$. To see the claim, observe that

$$
\frac{w_{\ell}(S)}{Z_{\ell}} = \frac{\prod_{g \in S} \frac{1}{x_g} \prod_{h \in \text{Ver}(\text{List}(\ell)) \setminus S} \left(1 - \frac{1}{x_h}\right)}{\prod_{f \in \text{Ver}(\text{List}(\ell))} (1 - \frac{1}{x_f})} = \frac{\prod_{g \in S} \frac{1}{x_g}}{\prod_{g \in S} (1 - \frac{1}{x_g})} = Q(S).
$$

Therefore, each $S \in \text{List}'(\ell)$ receives probability equal to

$$
\frac{w_{\ell}(S)}{Z_{\ell}} = \frac{Q(S)Z_{\ell}}{Q(B)Z_{\ell}} = \frac{Q(S)}{\sum_{B \in \text{List}'(\ell)} Q(B)}.
$$

**Lemma 2.** For any versioned witness forest $\phi$ with set of root-labels $R_\phi$, the branching process described earlier with input $|R_\phi|$ produces $\phi$ with probability

$$
p_\phi = \left(\frac{m}{|R_\phi|}\right) Q(R_\phi) \prod_{v \in \phi} \left[ (x_v - 1) \sum_{S \in \text{List}'(v)} Q(S) \right]^{-1}.
$$
Proof. Let $|R_\phi| = r$ and, for each node $v$ of $\phi$, let $N(v)$ denote the set of labels of its children. By Equation (14),

$$
\frac{1}{p_\phi} = \binom{m}{r} \prod_{v \in \phi} \frac{\sum_{S \in \text{List}'(v)} Q(S)}{Q(N(v))} = \binom{m}{r} \frac{\prod_{v \in \phi} \sum_{S \in \text{List}'(v)} Q(S)}{\prod_{v \in \phi \setminus R_\phi} \frac{1}{x_v - 1}}
$$

$$
= \binom{m}{r} Q(R_\phi) \prod_{v \in \phi} \left[ (x_v - 1) \sum_{S \in \text{List}'(v)} Q(S) \right]. \quad \square
$$

Let $\text{VWF}(r, t)$ denote the set of versioned witness forests with $r$ unversioned roots and exactly $t$ nodes. Since $\sum_{\phi \in \text{VWF}(r, t)} p_\phi \leq 1$, it follows that $|\text{VWF}(r, t)| \leq \min_{\phi \in \text{VWF}(r, t)} p_\phi$, which, by Lemma 2, equals

$$
\binom{m}{r} \max_{\phi \in \text{VWF}(r, t)} \left\{ Q(R_\phi) \prod_{v \in \phi} \left[ (x_v - 1) \sum_{S \in \text{List}'(v)} Q(S) \right] \right\}. \quad (15)
$$

Since for every $f \in F$, each $S \in \text{List}(f)$ gives rise to $\prod_{g \in S} Z/A_g$ sets in $\text{List}'(f)$, we get Equation (16). Setting $\mu_f = Z / A_f(x_f - 1) > 0$ in Equation (16) and recalling the definition of $\theta_f$ gives Equation (17).

$$
\prod_{v \in \phi} \left[ (x_v - 1) \sum_{S \in \text{List}'(v)} Q(S) \right] \leq \left( \max_{f \in F} \left[ (x_f - 1) \sum_{S \in \text{List}'(f)} Q(S) \right] \right)^t
$$

$$
= \left( \max_{f \in F} \left[ (x_f - 1) \sum_{S \in \text{List}'(f) g \in S} A_g(x_g - 1) \right] \right)^t
$$

$$
= \left( Z \max_{f \in F} \theta_f \right)^t. \quad (16)
$$

Substituting Equation (17) into Equation (15) yields

$$
|\text{VWF}(r, t)| \leq (\theta Z^t)^t \binom{m}{r} \max_{\phi \in \text{VWF}(r, t)} Q(R_\phi).
$$

To conclude, let $\gamma = \max_{f \in F} Z/A_f$ and let $\text{Roots}(r)$ denote the $r$-subsets in $\text{Roots}$. Assigning all possible version combinations to the roots of each forest in $\text{VWF}(r, t)$ and recalling Equation (13), we see that

$$
\frac{|\text{VerBad}(t)|}{Z^t} \leq \theta^t \left( \Omega \sum_{r=0}^{m} \binom{m}{r} \gamma^r \max_{S \in \text{Roots}(r)} \prod_{f \in S} \frac{\mu_f A_f}{Z} \right).
$$

Recalling that every versioned $t$-trajectory has probability at most $1/Z^t$, we see that the binary logarithm of the probability that the walk does not encounter a flawless
state within $t$ steps is at most $t \log_2 \theta + T_0$, where

$$T_0 = \log_2 |\Omega| + \log_2 \left( \sum_{r=0}^{m} \binom{m}{r} \gamma^r \max_{S \in \text{Roots}(S)} \prod_{f \in S} \frac{\mu_f A_f}{Z} \right)$$

$$\leq \log_2 |\Omega| + \log_2 \left( \sum_{r=0}^{m} \binom{m}{r} \left( \max_{f \in F} (\mu_f A_f) \right)^r \right)$$

$$= \log_2 |\Omega| + \max_{S \in \text{Roots}} |S| \cdot \log_2 \left( 1 + \frac{\max_{f \in F} (\mu_f A_f)}{\min_{f \in F} A_f} \right).$$

Therefore, if $t = (T_0 + s)/\log_2(1/\theta) \leq (T_0 + s)/\delta$, the probability that the random walk on $D^*$, and therefore on $D$, does not reach a flawless state within $t$ steps is at most $2^{-s}$.

8. A FIRST APPLICATION: HAMILTON CYCLES IN HYPERGRAPHS

8.1. Preliminaries

An (edge) coloring of a hypergraph $H(V, E)$ is a function $\phi : E \rightarrow \mathbb{N}$ assigning natural numbers (colors) to the edges of $H$. A hypergraph $H$ together with a given coloring $\phi$ will be dubbed a colored hypergraph. We will say that $e_1 \neq e_2 \in E$ are adjacent if $e_1 \cap e_2 \neq \emptyset$. A subhypergraph $S$ of a colored hypergraph $H$ is said to be properly colored if every two adjacent edges of $S$ receive different colors. If, further, every edge of $S$ receives a different color, that is, if $\phi$ is injective on $S$, we will say that $S$ is rainbow.

To promote the presence of properly colored and rainbow subhypergraphs, we introduce the following restrictions on hypergraph colorings. For a coloring $\phi$ and a color $i \in \mathbb{N}$, let $H_\phi^i = H[\phi^{-1}(i)]$ denote the hypergraph induced by the edges of color $i$ in $\phi$. We say that $\phi$ is $r$-degree bounded if $H_\phi^i$ has maximum degree at most $r$, for all $i \in \mathbb{N}$. If $H_\phi^i$ has at most $r$ edges, for all $i \in \mathbb{N}$, we say that $\phi$ is $r$-bounded.

We investigate the existence of properly colored and rainbow Hamilton cycles in colored $k$-uniform complete hypergraphs, $k \geq 3$. (A hypergraph is $k$-uniform if every edge has size $k$; it is complete if all $k$-element subsets of the vertices form edges.) For $1 \leq \ell < k$, an $\ell$-overlapping cycle is a $k$-uniform hypergraph in which, for some cyclic ordering of its vertices, every edge consists of $k$ consecutive vertices (in the cyclic ordering), and every two consecutive edges (in the natural ordering of the edges induced by the ordering of the vertices) share exactly $\ell$ vertices. Thus, the number of edges in an $\ell$-overlapping cycle with $s$ vertices is $s/(k-\ell)$. The two extreme cases $\ell = 1$ and $\ell = k-1$ are referred to as, respectively, loose and tight cycles.

Remark 6. A tight cycle on $s$ vertices contains an $\ell$-overlapping cycle on the same vertex set (with the same cyclic ordering), whenever $k - \ell$ divides $s$.

Given a $k$-uniform hypergraph $H$ on $n$ vertices where $k - \ell$ divides $n$, an $\ell$-overlapping cycle is called Hamilton if it goes through every vertex of $H$, that is, if $s = n$. We denote such a Hamilton cycle by $C_n^{(k)}(\ell)$. Let $K_n^{(k)}$ denote the complete $k$-uniform hypergraph on $n$ vertices.

Dudek et al. [2012] proved the following.

Theorem 1 ([DUDEK ET AL. 2012]). For every $1 \leq \ell < k$, there is a constant $c = c(k, \ell)$ such that if $n$ is sufficiently large and $k - \ell$ divides $n$, then any $cn^{k-\ell}$-bounded coloring of $K_n^{(k)}$ contains a rainbow copy of $C_n^{(k)}(\ell)$.
Theorem 2 ([Dudek et al. 2012]). For every 1 ≤ ℓ < k, there is a constant d = d(k, ℓ) such that if n is sufficiently large and k − ℓ divides n, then any \(dn^{k−ℓ}\)-degree bounded coloring of \(K_n^{(k)}\) contains a properly colored copy of \(C_n^{(k)(ℓ)}\).

Dudek and Ferrara [2015] strengthened Theorems 1 and 2 as follows. Say that a coloring is \((a, r)\)-bounded if, for each color \(i\), every set of \(a\) vertices is contained in at most \(r\) edges of color \(i\). An \(r\)-degree bounded coloring is, thus, \((1, r)\)-bounded and an \(r\)-bounded coloring is \((0, r)\)-bounded (as it has at most \(r\) edges of color \(i\)). Thus, Theorem 1 follows from Theorem 3 since, for every 1 ≤ ℓ ≤ k, every \(cn^{k−ℓ}\)-bounded coloring is both \((0, cn^{k−1})\)-bounded and \((ℓ, cn^{k−ℓ})\)-bounded. Similarly, Theorem 2 follows from Theorem 4 since, for every 1 ≤ ℓ ≤ k, every \(dn^{k−ℓ}\)-degree bounded coloring is \((ℓ, dn^{k−ℓ})\)-bounded.

Theorem 3 ([Dudek and Ferrara 2015]). For every 1 ≤ ℓ < k, there is a constant \(c = c(k, ℓ)\) such that if \(n\) is sufficiently large and \(k − ℓ\) divides \(n\), then any \((ℓ, cn^{k−ℓ})\)-bounded coloring of \(K_n^{(k)}\) that is \((0, cn^{k−1})\)-bounded contains a rainbow copy of \(C_n^{(k)(ℓ)}\).

Theorem 4 ([Dudek and Ferrara 2015]). For every 1 ≤ ℓ < k there is a constant \(d = d(k, ℓ)\) such that if \(n\) is sufficiently large and \(k − ℓ\) divides \(n\), then any \((ℓ, dn^{k−ℓ})\)-bounded coloring of \(K_n^{(k)}\) contains a properly colored copy of \(C_n^{(k)(ℓ)}\).

We make Theorems 3 and 4 constructive while also improving the constants from Dudek and Ferrara [2015].

Theorem 5. The Hamilton cycles guaranteed by Theorems 3 and 4 can be found in time \(O(n^{2k})\).

8.2. Proof of Theorem 5
We will use the following proposition, which can be derived easily by synthesizing results from Dudek et al. [2012].

Proposition 3. Fix 1 ≤ ℓ < k. Let \(\{e, f\}\) be any pair of edges of \(K_n^{(k)}\) with \(|e \cap f| = α ≤ ℓ\). Let \(X\) be any set of pairs \(\{g, h\}\) of edges of \(K_n^{(k)}\) satisfying \((e \cup f) ∩ (g \cup h) = \emptyset\).

Let \(C(X)\) be the set of all copies \(C\) of \(C_n^{(k)(k − 1)}\) in \(K_n^{(k)}\) such that \(\{g, h\} ∉ C\) for all \(\{g, h\} ∈ X\).

Let \(C_{e, f}(X) = \{C ∈ C(X) : \{e, f\} ⊂ C\}\).

There is \(δ = δ(k, ℓ) > 0\) such that, if \(C_{e, f}(X) \neq \emptyset\), one can find a disjoint family \(\{S_C : C ∈ C_{e, f}(X)\}\) of sets of copies of \(C_n^{(k)(k − 1)}\) from \(C(X)\) (indexed by the copies \(C ∈ C_{e, f}(X)\)) such that, for all \(C ∈ C_{e, f}(X)\):

1. \(S_C \cap C_{e, f}(X) = \emptyset\).
2. \(|S_C| ≥ δn^{2k−2}, if α = 0\).
3. \(|S_C| ≥ δn^{2k−α−1}, if 1 ≤ α ≤ ℓ\).

Furthermore, a uniformly random element of each set \(S_C\) can be sampled in time \(O(n^{2k})\).

Constructive Proof of Theorem 3. Fix 1 ≤ ℓ < k and let \(ϕ\) be a coloring of \(K_n^{(k)}\). Define the set \(M\) to consist of all pairs of edges that have the same color and share at most \(ℓ\) vertices, that is,

\[M = \{(e_1, e_2) : e_1, e_2 ∈ K_n^{(k)}, ϕ(e_1) = ϕ(e_2), and |e_1 \cap e_2| ≤ ℓ\}\]

Let \(Ω\) be the set of copies of \(C_n^{(k)(k − 1)}\) in \(K_n^{(k)}\). For each pair of edges \(\{e, f\} ∈ M\), we define the flaw

\[F_{e, f} = \{C ⊂ K_n^{(k)} : C ∼ C_n^{(k)(k − 1)} and \{e, f\} ⊂ C\}\].

Journal of the ACM, Vol. 63, No. 3, Article 22, Publication date: July 2016.
That is, \( F_{e,f} \) consists of all tight Hamilton cycles that contain both \( e \) and \( f \) (and are thus improperly colored since \( \phi(e) = \phi(f) \)). A flawless \( C \in \Omega \) is, thus, a tight Hamilton cycle whose edges have distinct colors. Since \( k - \ell \) divides \( n \), Remark 6 implies that any such cycle \( C \) contains a rainbow copy of \( C_n^{(k)}(\ell) \).

Having defined flaws, we now need to define actions for each flaw. To that end, for each pair \( \{e, f\} \in M \) and each integer \( 0 \leq \alpha \leq \ell \), we define

\[
Y_{e,f}(\alpha) = \{ \{e', f'\} \in M : \{e', f'\} \neq \{e, f\}, |e' \cap f'| = \alpha \text{ and } (e \cup f) \cap (e' \cup f') \neq \emptyset \}.
\]

Let

\[
Y_{e,f} = \bigcup_{\alpha=0}^{\ell} Y_{e,f}(\alpha) \quad \text{and} \quad X_{e,f} = M \setminus (Y_{e,f} \cup \{e, f\}).
\]

For each Hamilton cycle \( C \in \Omega \), for each pair of edges \( \{e, f\} \in M \) such that flaw \( F_{e,f} \) is present in \( C \), we invoke Proposition 3 with \( e, f \), and \( X = X_{e,f} \). Let \( S_C \) be the set of Hamilton cycles guaranteed by Proposition 3. We let \( A(F_{e,f}, C) = S_C \). To lighten notation, we let \( A_{e,f} := A_{F_{e,f}} = \min_{C \in \Omega} |A(F_{e,f}, C)| \). By Proposition 3, we thus have that:

— \( D \) is atomic since, for each flaw, we have a disjoint family of sets of cycles (actions).
— If \( |e \cap f| = \alpha \), then

\[
A_{e,f} = \begin{cases} 
\frac{\delta n^{2k-2}}{2ckn^{2k-1}} & \text{if } \alpha = 0, \\
\frac{\delta n^{2k-1-\alpha}}{2ck^{\ell+1}n^{2k-2-\alpha}} & \text{if } 1 \leq \alpha \leq \ell.
\end{cases}
\]

— If \( F_{g,h} \in \Gamma(F_{e,f}) \), then \( \{g, h\} \in Y_{e,f} \) since \( \{g, h\} \notin X_{e,f} \) and \( \{g, h\} \notin \{e, f\} \)

To bound \( |Y_{e,f}| \), we use the following fact, established in Dudek and Ferrara [2015]. For every \( c > 0 \), if \( \phi \) is \((\ell, cn^{k-\ell})\)-bounded and \((0, cn^{k-1})\)-bounded, then there exists \( n_0 = n_0(c) \) such that, for all \( n \geq n_0 \),

\[
\max_{\{e, f\} \in M} |Y_{e,f}(\alpha)| \leq \begin{cases} 
2ckn^{2k-2} & \text{if } \alpha = 0, \\
2ck^{\ell+1}n^{2k-2} & \text{if } 1 \leq \alpha \leq \ell.
\end{cases}
\]

Therefore, for each pair of edges \( \{e, f\} \in M \) with \( |e \cap f| = \alpha \),

\[
\sum_{F_{g,h} \in \Gamma(F_{e,f})} \frac{1}{A_{g,h}} = \sum_{\alpha=0}^{\ell} \sum_{\{g, h\} \in Y_{e,f}(\alpha)} \frac{1}{A_{g,h}} \leq \frac{2ckn^{2k-2}}{\delta n^{2k-2}} + \sum_{\alpha=1}^{\ell} \frac{2ck^{\ell+1}n^{2k-2-\alpha}}{\delta n^{2k-2-\alpha}} = \frac{2ck}{\delta} (1 + \ell k^\ell).
\]

Thus, for any \( c < 0.5(2ek(1 + \ell k^\ell))^{-1} \), the right-hand side is less than \( 1/e \).

By Theorem 1, the uniform random walk on \( D \) terminates after \( O(|M| + \log |\Omega|) \) steps with high probability, where \( \log_2 |\Omega| \leq \log_2 \left( \frac{n}{k} \right)^n \leq nk \log_2 n \) and \( |M| \leq n^{2k} \). As in each step we need \( O(n^{2k}) \) time to find the greatest flaw and \( O(n^{2k}) \) time to choose an action for it, we have proven the theorem. \( \Box \)

**Constructive Proof of Theorem 4.** Fix \( 1 \leq k < k \) and let \( \phi \) be a coloring of \( K_n^{(k)} \). The proof goes along the lines of the proof of Theorem 3, except that we slightly modify the definition of the set \( M \) in that it contains no pair of disjoint edges, that is,

\[
M = \{ \{e_1, e_2\} : e_1, e_2 \in K_n^{(k)}, 1 \leq |e_1 \cap e_2| \leq \ell, \text{ and } \phi(e_1) = \phi(e_2) \}.
\]

We change \( M \) in this way since a properly colored cycle may contain nonadjacent edges of the same color. As before, for each pair of edges \( \{e, f\} \in M \), define the flaw

\[
F_{e,f} = \{ C \subset K_n^{(k)} : C \sim C_n^{(k)}(k - 1) \text{ and } \{e, f\} \subset C \}.
\]
A flawless $C \in \Omega$ is, thus, a tight Hamilton cycle such that, for every pair of its edges $e$ and $f$ with $1 \leq |e \cap f| \leq \ell$, we have that $\phi(e) \neq \phi(f)$. Since $k - \ell$ divides $n$, once again Remark 6 implies that any such cycle $C$ contains a properly colored copy of $C_n^{(k)}(\ell)$.

We define the sets $Y_{e,f}(\alpha)$ only for $\alpha \in [\ell]$ and, consequently, $Y_{e,f} = \bigcup_{\alpha=1}^{\ell} Y_{e,f}(\alpha)$. An argument identical to that in the proof of Theorem 3 shows that if $d = \delta(2e\ell k^{\ell+1})^{-1}$, Theorem 1 applies. \[\square\]

9. RAINBOW SUBGRAPHS OF COMPLETE GRAPHS

In an edge-colored graph $G = (V, E)$, say that $S \subseteq E$ is rainbow if its elements have distinct colors.

9.1. Matchings

**Theorem 1.** For any $C > 2e$, given any edge-coloring of the complete graph on $2n$ vertices in which each color appears on at most $n/C$ edges, a rainbow perfect matching can be found in $O(n^4 \log n)$ time.

**Proof.** Let $\phi$ be any edge-coloring of $K_{2n}$ in which each color appears on at most $q$ edges. Let $P = P(\phi)$ be the set of all pairs of vertex-disjoint edges with the same color in $\phi$, that is, $P = \{\{e_1, e_2\} : \phi(e_1) = \phi(e_2)\}$. Let $\Omega$ be the set of all perfect matchings of $K_{2n}$. For each $\{e_i, e_j\} \in P$, let

$$f_{i,j} = \{M \in \Omega : \{e_i, e_j\} \subset M\}.$$

Thus, an element of $\Omega$ is flawless if and only if it is a rainbow perfect matching.

To address the flaw induced by edges $e_i, e_j \in M$, we select two other edges $e_k, e_\ell \in M$ and in each of the two edge-pairs $\{e_i, e_k\}$ and $\{e_j, e_\ell\}$, we select one of the other two matchings. More precisely, let $f = \{(v_1, v_2), (v_3, v_4)\} \in P$ and assume, without loss of generality, that $v_1 > v_2, v_3 > v_4$, and $v_1 > v_3$. For $M \in f$, the set $A(f, M)$ consists of all possible outputs of $\text{Switch}(M, \{v_1, v_2\}, \{v_3, v_4\})$.

**Algorithm 4:** Switch$(M, \{v_1, v_2\}, \{v_3, v_4\})$

1. Let $v_1$ be any vertex other than $\{v_1, v_2, v_3, v_4\}$.
2. Let $v_2$ be the vertex such that $(u_1, u_2) \in M$.
3. Let $v_3$ be any vertex other than $\{v_1, v_2, v_3, v_4, u_1, u_2\}$.
4. Let $u_4$ be the vertex such that $(u_3, u_4) \in M$.
5. Output $M' \in \Omega$ by removing from $M$ edges $(v_1, v_2), (u_1, u_2), (v_3, v_4), (u_3, u_4)$, and adding edges $(v_1, u_1), (v_2, u_2), (v_3, u_3), (v_4, u_4)$.

Enumerating the choices in Steps 1 and 3, we see that $|A(f, M)| = (2n - 4)(2n - 6)$. On the other hand, each of the four edges inserted by each action of each set $A(f, \cdot)$ has exactly one vertex in $V = \{v_1, \ldots, v_4\}$ and one vertex outside $V$, and can only form a flaw along with an edge having the same color as itself. Therefore, $|\Gamma(f_{i,j})| \leq 4(2n - 4)(q - 1)$ for every $i, j$.

Let $D$ be the directed graph on $\Omega$ corresponding to the sets of actions described. To prove atomicity, consider any arc $M \xrightarrow{f} M'$. Adding to $M'$ the two edges defining $f$ yields two edge-disjoint paths of length 3. Considering the 4-cycle closing each path, $\bar{M}$ results by taking in each cycle the 2-matching containing the edges of the flaw. Let $A_{i,j} := A_{f_{i,j}} = \min_{M \in f} |A(f, M)|$. To conclude, we note that, for every $C > 2e$, there
exists $\delta = \delta(C)$, such that, for all $n \geq n_0(C)$,
\[\sum_{f_k \in \Pi(f_1)} \frac{1}{A_{k, \ell}} \leq \frac{4(2n - 4)(q - 1)}{(2n - 4)(2n - 6)} = \frac{2(q - 1)}{n - 3} \leq \frac{1}{e - \delta}.
\]

To bound the runtime, let $q_i$ be the number of edges with color $i$. Trivially, $|P| \leq \sum_i (\frac{q}{n}) < n^4$, and it is not hard to see that, in fact, $|P| < n^3$. At the same time, $|\Omega| = (2n - 1)!! = (2n - 1)(2n - 3) \cdots 1$, implying that $\log_2 |\Omega| = O(n \log n)$. By Theorem 1, the uniform random walk on $D$ terminates after $O(|P| + \log_2 |\Omega|)$ steps with high probability. By fixing an arbitrary ordering of the colors and after each step sorting the edges in the current matching according to color, we can trivially find the greatest flaw present in time $O(n \log n)$ (flaws are ordered first by color, then in lexicographic order of the four vertices involved).

**10. COLOR-BLIND INDEX OF GRAPHS: WHEN DIRECTED CAUSALITY MATTERS**

### 10.1. Preliminaries

Let $\phi : E(G) \to [k]$ be an edge-coloring, not necessarily proper, of a graph $G(V, E)$. For each vertex $v \in V$, let $c(v) = (a_1, \ldots, a_k)$, where $a_i = |\{u : (u, v) \in E, c((u, v)) = i\}|$ for $i \in [k]$. We say that $\phi$ distinguishes neighbors by multisets if $c$ is a proper vertex-coloring of $V$ and denote by $ndm(G)$ the smallest $k$ for which such $c$ exists. Clearly, if $G$ contains $K_2$ as a connected component, no such edge-coloring exists. Addario-Berry et al. [2005] proved that as long as that is not the case, $ndm(G) \leq 4$.

Kalinowski et al. [2013] introduced a fascinating twist to this concept that captures color-blindness. A color-blind person looking at two green edges and one red edge sees two edges of the same color and one edge of another color. Their view would remain the same if, instead, we had two red edges and one green. If we reorder the sequence $c(v) = (a_1, a_2, \ldots, a_k)$ nondecreasingly, we obtain a sequence $p(v) = (d_1, \ldots, d_k)$, called the palette of vertex $v$. (Thus, there is a bijection between the set of all possible palettes of a vertex $v$ of degree $d$ and the set of all partitions of the integer $d$ into at most $k$ parts.) We say that a color-blind person can distinguish neighbors if $p(u) \neq p(v)$ for every edge $(u, v) \in E$, that is, if $p$ is a proper coloring of the vertices of $G$. The smallest possible number $k$ for which such an edge-coloring exists is called the color-blind index of a graph $G$ and is denoted by $\text{dal}(G)$, the notation referring to the English chemist John Dalton, who was the first scientist to take academic interest in the subject of color blindness.

It has to be noted that there are infinitely many graphs, for example, odd cycles, for which the color-blind index is not defined. In Kalinowski et al. [2013], it was conjectured that there exists a number $K$ such that $\text{dal}(G) \leq K$ for every graph $G$ for which $\text{dal}(G)$ is defined. The authors prove this conjecture for complete graphs, regular bipartite graphs, regular graphs of sufficiently large degree, and graphs with bounded ratio $\Delta(G)/\delta(G)$.

**Theorem 1** ([Kalinowski et al. 2013]). For every $R \geq 1$, there exists $\delta_0 = \delta_0(R)$ such that if $\delta(G) \geq \delta_0$ and $\Delta(G) \leq R\delta(G)$, then $\text{dal}(G) \leq 6$.

Theorem 1, covering both regular and irregular graphs, is proven in Kalinowski et al. [2013] by applying the lopsided LLL with a directed lopsidependency graph. Thus, the result does not fit either the variable framework of Moser and Tardos [2010] or the permutation setting of Harris and Srinivasan [2014].

**Theorem 2.** The colorings guaranteed by Theorem 1 can be found in $O(|E(G)|(1 + \Delta(G) - \delta(G)))$ time.
10.2. Proof
Let $\Omega$ be the set of all edge-colorings, not necessarily valid, of $G(V, E)$ with 6 colors. Let $S_v$ denote the set of edges incident to a vertex $v \in V$, and for $\{u, v\} \in E$, let $S_{u,v} = S_u \cup S_v$. Fix an arbitrary ordering of $V$. For each edge $\{u, v\}$, where $u < v$ and $d(u) = d(v)$, we define a set of flaws as follows.

Let $\text{Bad}(u, v)$ be the set of all edge colorings of $G$ with $p(u) = p(v)$. Partition $\text{Bad}(u, v)$ into equivalence classes, forming a partition $P$, where two colorings are equivalent if they agree on the coloring of $S_u \setminus \{u, v\}$. Further partition each class $C \in P$ into equivalence classes, forming a partition $Q(C)$, where two colorings in $C$ belong in the same equivalence class if they agree on the coloring of $S_v$.

We claim that if $d(u) = d(v) = d$, then, for each $C \in P$, the size of $Q(C)$, that is, the number of equivalence classes in $Q(C)$, is at most

$$f(d) := \max_{d_1 + \ldots + d_6 = d} \left(\begin{array}{c} d \\ d_1, d_2, \ldots, d_6 \end{array}\right) 6!.$$  \hspace{1cm} (18)

To see this, observe that, for any coloring of $S_u \setminus \{u, v\}$, there exist numbers $d_1 \geq d_2 \geq \cdots \geq d_6$ summing to $d$ such that $p(u) = p(v)$ implies that $p(v) = (d_1, d_2, \ldots, d_6)$. Therefore, the number of elements in $Q(C)$ is bounded by the number of ways to partition the $d$ edges in $S_v$ into sets of sizes $d_1, \ldots, d_6$ times the $6!$ ways of assigning distinct colors to the sets. Finally, it is not hard to see that Equation (18) is maximized when $|d_i - d_j| \leq 1$ for all $i, j \in [6]$, implying that

$$f(d) = \max_{d_1 + \ldots + d_6 = d} \left(\begin{array}{c} d \\ d_1, d_2, \ldots, d_6 \end{array}\right) 6! \leq 6^d \frac{27\sqrt{2}}{\pi d^{5/2}} 6! < 1572 \frac{6^d}{d^{5/2}}.$$  \hspace{1cm} (19)

For each $C \in P$, consider an arbitrary ordering for the members of $Q(C)$ and let $Z_i(C)$, $i \in [f(d)]$, be the $i$th member of $Q(C)$. For each $i \in [f(d)]$, define the flaw $f^{i}_{u,v}$ as:

$$f^{i}_{u,v} = \bigcup_{C \in P} Z_i(C).$$

Thus, a flawless element is an element for which there is no edge $(u,v)$ such that $p(u) = p(v)$, as we wanted. For each flaw $f^{i}_{u,v}$, where $u < v$, for each $\phi \in f^{i}_{u,v}$, the set of actions $A(f^{i}_{u,v}, \phi) \subseteq \Omega$ consists of all possible recolorings of $S_v$ in $\phi$. Thus, $|A(f^{i}_{u,v}, \phi)| = 6^d$ for all $u, v, i$.

Let $D$ be the directed graph on $\Omega$ corresponding to these actions. To establish the atomicity of $D$, it suffices to show that for every transition $\phi \xrightarrow{f^{a}_{u,v}} \phi'$, where $u < v$, if we are given $\phi'$ and $f^{a}_{u,v}$, we can reconstruct $\phi$. To see this, at first notice that $\phi$ and $\phi'$ differ only in the coloring of $S_v$. Therefore, $\phi'$ implies the coloring $C$ of $S_u \setminus \{u, v\}$ in $\phi$, while the integer $i$ implies $Z_i(C)$ and, therefore, the coloring of $S_v$.

Fix $\{u, v\} \in E$ with $u < v$ and $d(u) = d(v)$ and let $M(u, v) = \bigcup_{e \in S_e} S_e$. That is, $M(u, v)$ is the set of edges that are adjacent to $u$ or $v$, or to edges adjacent to $v$. Observe that any action taken at a state $\phi$ to address flaw $f^{a}_{u,v}$ only introduces flaws that are associated with edges in $M(u, v)$. To see this, notice that when we recolor an edge, we only introduce flaws associated with edges adjacent to it.

Recall by our earlier discussion that for any edge whose endpoints have degree $d$, the number of flaws associated with it is bounded by $f(d)$ and that for any such flaw $g$, we have $A_g := \min_{\phi \in g} |A(g, \phi)| = 6^d$. Therefore, for any such flaw $f^{a}_{u,v}$ (thus, for every
flaw),
\[
\sum_{g \in \Gamma(f_{i,j,v,j})} 1 < \sum_{e \in M(u,v)} f(e) < |M(u,v)| \cdot \frac{1572}{\delta(G)^{5/2}} \leq 1572 \frac{\Delta(G)^2}{\delta(G)^{5/2}}.
\] (20)

From Equation (20), we see that if \(\delta(G) > \delta_0 = (1572eR^2)^2\), then the condition of Theorem 1 holds. Regarding the runtime, we note that \(\log_2 |\Omega| = \log_2 6^{|E|} = O(|E|)\) while, clearly, \(|U(\sigma_1)| \leq |E|\) since, in each state, each edge can “give rise” to at most one flaw.

**11. LATIN TRANSVERSALS - A CASE OF DENSE NEIGHBORHOODS**

Let \(M\) be an \(n \times n\) matrix whose entries come from a set of colors \(C\). A Latin Transversal of \(M\) is a permutation \(\pi \in S_n\) such that the entries \([M(i, \pi(i))]_{i=1}^n\) have distinct colors, that is, a selection of \(n\) entries in distinct rows and columns such that no two elements have the same color.

**Theorem 1.** If each color \(c \in C\) appears at most \(\Delta \leq \frac{2572}{256} n\) times in \(M\), then the Recursive Walk will find a Latin Transversal of \(M\) in \(O(n \log n)\) steps with high probability.

**Proof.** Let \(M\) be any matrix in which each color appears at most \(\Delta\) times and let \(\Omega = S_n\) be the set of all permutations of \([n]\). Let \(P = P(M)\) be the set of all quadruples \((i, j, i', j')\) such that \(M(i, j) = M(i', j')\). For each quadruple \((i, j, i', j') \in P\), let
\[
f_{i,j,i',j'} = \{ \pi \in \Omega : \pi(i) = j \text{ and } \pi(i') = j' \}.
\]
Thus, an element of \(\Omega\) is flawless if and only if it is a Latin Transversal of \(M\).

To address the flaw induced by a pair of entries \((i, j), (i', j')\) of \(M\) in an element \(\pi \in \Omega\), we select two other entries \((\alpha, \beta), (\alpha', \beta')\), also selected by \(\pi\), and replace the four entries \((i, j), (i', j'), (\alpha, \beta), (\alpha', \beta')\) with the four entries \((i, \beta), (i', \beta'), (\alpha, j), (\alpha', j')\). More precisely, for \(\pi \in f = f_{i,j,i',j'}\), the set \(A(f, \pi)\) consists of all possible outputs of Switch\((\pi, i, j, i', j')\).

**ALGORITHM 5:** Switch\((\pi, i, j, i', j')\)

1. Let \(\alpha\) be any element of \([n]\). Let \(\beta = \pi(\alpha)\).
2. Let \(\alpha' \neq \alpha\) be any element of \([n]\). Let \(\beta' = \pi(\alpha')\).
3. Modify \(\pi\) to \(\beta\) by the following “switch”: \(\rho(i) = \beta, \rho(i') = \beta', \rho(\alpha) = j, \rho(\alpha') = j'\).

To prove atomicity, consider any action \(\frac{f_{i,j,i',j'}}{\delta(G)} \rho\). Suppose that \(\rho(i) = \beta, \rho(i') = \beta', \rho^{-1}(j) = \alpha, \text{ and } \rho^{-1}(j') = \alpha'\). Given \(\rho\) and \((i, j, i', j')\), we see that the image of every element under \(\rho\) other than \(i, i', \alpha, \alpha'\) is the same as under \(\rho\), while \(\pi(i) = j, \pi(i') = j'\), \(\pi(\alpha) = \beta\) and \(\pi(\alpha') = \beta'\).

Enumerating the choices in Steps 1 and 2, we see that \(|A(f, \pi)| = n(n-1)\). Let us now consider the form of the causality graph \(C\). It is not hard to see that if \(f \to g\) exists in \(C\), then \(g \to f\) exists as well; thus, we will think of the undirected version \(G\) of \(C\). Two flaws \(f_{i,j,i',j'}\) and \(f_{p,q,p',q'}\) are adjacent in \(G\) if and only if \([i, i'] \setminus \{p, p'\} \neq \emptyset\) or \([j, j'] \setminus \{q, q'\} \neq \emptyset\).

Thus, each flaw \(f_{i,j,i',j'}\) is adjacent to four types of flaws, corresponding to the four new entries \((i, j), (i', j'), (\alpha, i), (\alpha', j')\). The maximum degree of \(G\) is at most \(4n(\Delta - 1)\) since, for a fixed \((i, j, i', j')\), we can choose \((s, t)\) with \(s \in \{i, i'\}\) or \(t \in \{j, j'\}\) in \(4n\) different ways and, as there are at most \(\Delta\) entries of \(M\) with any given color, once \((s, t)\) has been chosen, there are at most \(\Delta - 1\) choices for \((s', t')\) such that \(M(s, t) = M(s', t')\). Thus, the set of vertices in \(\Gamma(f_{i,j,i',j'})\) is the union of four subsets, each of cardinality at most \(n(\Delta - 1)\), where crucially the vertices in each subset form a clique.

---

Journal of the ACM, Vol. 63, No. 3, Article 22, Publication date: July 2016.
Setting $\mu_f = \mu$ for each flaw $f$, condition (7) of Theorem 4 becomes

$$n(n - 1) \geq \mu^{-1} \sum_{i=0}^{4} \binom{4}{i} (n(\Delta - 1))^i = \mu^{-1} (1 + \mu n(\Delta - 1))^4.$$  \hfill (21)

It is easy to see that if $\mu = \frac{1}{3n^{\Delta-1}}$, then Equation (21) holds for all $\Delta \leq \frac{27}{256} n$.

To bound the runtime, note that, for every state $\sigma_1$, the largest independent subset of $U(\sigma_1)$ is of size $O(n)$ and that $\log_2 |\Omega| = \log_2 n! = \Theta(n \log n)$. \hfill $\Box$

ACKNOWLEDGMENTS

Dimitris Achlioptas thanks Christos Papadimitriou for introducing him to the Probabilistic Method a quarter century ago and Mario Szegedy for inspiration to avoid the lowest common denominator ($Z$).

REFERENCES

Dimitris Achlioptas and Fotis Iliopoulos. 2014. 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, October 18–21, 2014. IEEE Computer Society, 494–503. DOI: http://dx.doi.org/10.1109/FOCS.2014.59

Louigi Addario-Berry, Robert E. L. Aldred, Ketan Dalal, and Bruce A. Reed. 2005. Vertex colouring edge partitions. Journal of Combinatorial Theory, Series B 94, 2, 237–244. DOI: http://dx.doi.org/10.1016/j.jctb.2005.01.001

Noga Alon. 1991. A parallel algorithmic version of the local lemma. Random Structures & Algorithms 2, 4, 367–378.

József Beck. 1991. An algorithmic approach to the Lovász local lemma. I. Random Structures & Algorithms 2, 4 (1991), 343–365. DOI: http://dx.doi.org/10.1002/rsa.3240020402

Rodrigo Bissacot, Roberto Fernández, Aldo Procaccia, and Benedetto Scoppola. 2011. An improvement of the Lovász local lemma via cluster expansion. Combinatorics, Probability & Computing 20, 5, 709–719.

Karthekeyan Chandrasekaran, Navin Goyal, and Bernhard Haeupler. 2013. Deterministic algorithms for the Lovász local lemma. SIAM Journal on Computing 42, 6, 2132–2155. DOI: http://dx.doi.org/10.1137/100799642

Artur Czumaj and Christian Scheideler. 2000. Coloring nonuniform hypergraphs: A new algorithmic approach to the general Lovász local lemma. Random Structures & Algorithms 17, 3–4, 213–237.

Persi Diaconis. 1996. The cutoff phenomenon in finite Markov chains. Proceedings of the National Academy of Sciences 93, 4, 1659–1664.

Andrzej Dudek and Michael Ferrara. 2015. Extensions of results on rainbow Hamilton cycles in uniform hypergraphs. Graphs and Combinatorics 31, 3, 577–583. DOI: http://dx.doi.org/10.1007/s00373-013-1391-z

Andrzej Dudek, Alan M. Frieze, and Andrzej Rucinski. 2012. Rainbow Hamilton cycles in uniform hypergraphs. Electronic Journal of Combinatorics 19, 1, P46.

Paul Erdős and László Lovász. 1975. 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. North-Holland, Amsterdam, 609–627. Colloq. Math. Soc. János Bolyai, Vol. 10.

Paul Erdős and Joel Spencer. 1991. Lopsided Lovász local lemma and latin transversals. Discrete Applied Mathematics 30, 2–3, 151–154.

Bernhard Haeupler, Barna Saha, and Aravind Srinivasan. 2011. New constructive aspects of the Lovász local lemma. Journal of the ACM 58, 6, 28. DOI: http://dx.doi.org/10.1145/2049697.2049702

David G. Harris and Aravind Srinivasan. 2013. The Moser-Tardos framework with partial resampling. In 54th Annual IEEE Symposium on Foundations of Computer Science, (FOCS’13), Berkeley, CA. IEEE Computer Society. 469–478. DOI: http://dx.doi.org/10.1109/FOCS.2013.57

David G. Harris and Aravind Srinivasan. 2014. A constructive algorithm for the Lovász local lemma on permutations. In SODA. SIAM, 907–925.

Nicholas J. A. Harvey and Jan Vondrák. 2015. An algorithmic proof of the Lovász local lemma via resampling oracles. In IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS’15), Berkeley, CA. Venkatesan Guruswami (Ed.). IEEE Computer Society, 1327–1346. DOI: http://dx.doi.org/10.1109/FOCS.2015.85

Rafał Kalinowski, Monika Pilśniak, Jakub Przybyło, and Mariusz Woźniak. 2013. Can colour-blind distinguish colour palettes? The Electronic Journal of Combinatorics 20, 3, P23.

Journal of the ACM, Vol. 63, No. 3, Article 22, Publication date: July 2016.
Kashyap Kolipaka, Mario Szegedy, and Yixin Xu. 2012. A sharper local lemma with improved applications. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, Anupam Gupta, Klaus Jansen, José Rolim, and Rocco Servedio (Eds.). Lecture Notes in Computer Science, Vol. 7408. Springer, Berlin, 603–614. DOI: http://dx.doi.org/10.1007/978-3-642-32312-0_51

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

Linyuan Lu, Austin Mohr, and László Székely. 2013. Quest for negative dependency graphs. In *Recent Advances in Harmonic Analysis and Applications*. Springer, 243–258.

Austin Mohr. 2013. *Applications of the Lopsided Lovász Local Lemma Regarding Hypergraphs*. Ph.D. Dissertation. University of South Carolina, Columbia, SC.

Michael Molloy and Bruce Reed. 1999. Further algorithmic aspects of the local lemma. In *STOC'98*. ACM, New York, NY, 524–529.

Robin A. Moser. 2009. A constructive proof of the Lovász local lemma. In *Proceedings of the 2009 ACM International Symposium on Theory of Computing (STOC'09)*. ACM, New York, NY, 343–350.

Robin A. Moser and Gábor Tardos. 2010. A constructive proof of the general Lovász local lemma. *Journal of the ACM* 57, 2, Article 11, 15. DOI: http://dx.doi.org/10.1145/1667053.1667060

Christos H. Papadimitriou. 1991. On selecting a satisfying truth assignment. In *FOCS*. IEEE Computer Society, 163–169.

Wesley Pegden. 2011. Highly nonrepetitive sequences: Winning strategies from the local lemma. *Random Structures & Algorithms* 38, 1–2, 140–161.

Wesley Pegden. 2014. An extension of the Moser-Tardos algorithmic local lemma. *SIAM Journal on Discrete Mathematics* 28, 2, 911–917. DOI: http://dx.doi.org/10.1137/110828290

James B. Shearer. 1985. On a problem of Spencer. *Combinatorica* 5, 3, 241–245. DOI: http://dx.doi.org/10.1007/BF02579368

Joel Spencer. 1977. Asymptotic lower bounds for Ramsey functions. *Discrete Mathematics* 20, 0, 69–76. DOI: http://dx.doi.org/10.1016/0012-365X(77)90044-9

Aravind Srinivasan. 2008. Improved algorithmic versions of the Lovász local lemma. In *SODA*, Shang-Hua Teng (Ed.). SIAM, 611–620.

Mario Szegedy. 2013. The Lovász local lemma—a survey. In *CSR*, Lecture Notes in Computer Science, Andrei A. Bulatov and Arseny M. Shur (Eds.), Vol. 7913. Springer, 1–11.

Received April 2015; revised December 2015; accepted December 2015