SIAM J. COMPUT.
Vol. 48, No. 5, pp. 1583–1602

A LOCAL LEMMA FOR FOCUSED STOCHASTIC ALGORITHMS*

DIMITRIS ACHLIOPTAS†, FOTIS ILIOPOULOS‡, AND VLADIMIR KOLMOGOROV§

Abstract. We develop a framework for the rigorous analysis of focused stochastic local search algorithms. These algorithms search a state space by repeatedly selecting some constraint that is violated in the current state and moving to a random nearby state that addresses the violation, while (we hope) not introducing many new violations. An important class of focused local search algorithms with provable performance guarantees has recently arisen from algorithmizations of the Lovász local lemma (LLL), a nonconstructive tool for proving the existence of satisfying states by introducing a background measure on the state space. While powerful, the state transitions of algorithms in this class must be, in a precise sense, perfectly compatible with the background measure. In many applications this is a very restrictive requirement, and one needs to step outside the class. Here we introduce the notion of *measure distortion* and develop a framework for analyzing arbitrary focused stochastic local search algorithms, recovering LLL algorithmizations as the special case of no distortion. Our framework takes as input an arbitrary algorithm of such type and an arbitrary probability measure and shows how to use the measure as a yardstick of algorithmic progress, even for algorithms designed independently of the measure.

Key words. Lovász local lemma, Moser–Tardos algorithm, stochastic local search

AMS subject classifications. 68W20, 68Q87, 60C05

DOI. 10.1137/16M109332X

1. Introduction.

Let Ω be a large, finite set of objects, and let $F = \{f_1, f_2, \ldots, f_m\}$ be a collection of subsets of Ω. We will refer to each $f_i \in F$ as a *flaw* to express that all objects in f_i have negative feature $i \in [m]$. For example, for a CNF formula on n variables with clauses $c_1, c_2, \ldots, c_m$, for each clause c_i we can define $f_i \subseteq \{0,1\}^n$ as the truth assignments that violate c_i. Following linguistic rather than mathematical convention, we will say that flaw f is present in object σ if $f \ni \sigma$ and that $\sigma \in \Omega$ is *flawless* if no flaw is present in σ.

The Lovász local lemma (LLL)[1] is a nonconstructive tool for proving the *existence* of flawless objects by introducing a probability measure μ on Ω and bounding from below the probability of simultaneously avoiding all ("bad") events corresponding to the flaws in F. (Throughout, products devoid of factors equal 1.)

GENERAL LLL. *Given events $A_1, \ldots, A_m$, for each $i \in [m]$, let the set $D(i) \subseteq [m] \setminus \{i\}$ be such that if $S \subseteq [m] \setminus (D(i) \cup \{i\})$, then $\mu(A_i \mid \cap_{j \in S} \overline{A_j}) = \mu(A_i)$. If there*

*Received by the editors September 9, 2016; accepted for publication (in revised form) July 15, 2019; published electronically October 31, 2019. This paper is based on results that appeared in preliminary form in [2] and were improved, among other contributions, in [23].

https://doi.org/10.1137/16M109332X

Funding: The first author was supported by NSF grant CCF-1514128. The second author was supported by NSF grant CCF-1514434. The third author was supported by ERC grant agreement 616160 through the European Research Council under the Seventh Framework Programme (FP7/2007–2013).

†Department of Computer Science, University of California Santa Cruz, Santa Cruz, CA 95064 (optas@cs.ucsc.edu).

‡Department of Electrical Engineering and Computer Science, University of California Berkeley, Berkeley, CA 94720-1776 (fotis.iliopoulos@berkeley.edu).

§Institute of Science and Technology Austria, 3400 Klosterneuburg, Austria (vnk@ist.ac.at).

[1]Condition (1) is equivalent to the more common form requiring $\mu(A_i) \le x_i \prod_{j \in D(i)} (1 - x_j)$, for $x_i \in [0,1)$; see section 2.1. We prefer (1) because, as we will see, it is better suited for stating refinements of the condition.

exist $\{\psi_i\}_{i=1}^m > 0$ *such that for all* $i \in [m]$,

$$\frac{\mu(A_i)}{\psi_i} \sum_{S \subseteq \{i\} \cup D(i)} \prod_{j \in S} \psi_j \leq 1 \ , \tag{1}$$

then the probability that none of $A_1, \ldots, A_m$ *occurs is at least* $\prod_{i=1}^m 1/(1+\psi_i) > 0$.

Erdős and Spencer [12] noted that independence in the LLL can be replaced by *positive correlation*, yielding the stronger lopsided LLL. The difference is that each set $D(i)$ is replaced by a set $L(i) \subseteq [m] \setminus \{i\}$ such that if $S \subseteq [m] \setminus (L(i) \cup \{i\})$, then $\mu(A_i \mid \cap_{j \in S} \overline{A_j}) \leq \mu(A_i)$, i.e., "=" is replaced by "≤." Considering graphical properties of the graphs on $[m]$ induced by the relationships $D(\cdot)$ and $L(\cdot)$, we can show more permissive conditions, such as the cluster expansion [6], the left-handed [31], and Shearer's condition [34].

Moser [27], later joined by Tardos [28], showed in groundbreaking work that a simple algorithm can be used to make the general LLL constructive when μ is a product measure. Specifically, in the *variable setting* of [28], each event A_i is determined by a set of variables $\mathrm{vbl}(A_i)$ so that $j \in D(i)$ iff $\mathrm{vbl}(A_i) \cap \mathrm{vbl}(A_j) \neq \emptyset$. Moser and Tardos proved that in the variable setting, if (1) holds, then repeatedly selecting *any* occurring event A_i and resampling every variable in $\mathrm{vbl}(A_i)$ independently according to μ leads to a flawless object after a linear expected number of resamplings. Pegden [32] proved that this remains true under the weakening of (1) to the cluster expansion condition of Bissacot et al. [6]. Finally, Kolipaka and Szegedy [22] proved that the resampling algorithm actually works even under Shearer's tight condition [34]. In an orthogonal development, Harris and Srinivasan [18] were the first to make the LLL constructive outside the variable setting, giving an algorithmic LLL for the uniform measure on permutations.

Moser's original analysis of the resampling algorithm in the context of satisfiability [27] inspired a parallel line of works that formed the so-called entropy compression method; see, e.g., [11, 16, 13]. In these works, the set of objects Ω typically does not have product structure, and there is no measure μ and no general condition for algorithmic convergence. Instead, the fact that the algorithm under consideration must reach a flawless object (and thus terminate) is established by proving that the entropy of its trajectory grows more slowly than the rate at which it consumes randomness. The rate comparison is done in each case via a problem-specific counting argument.

A common feature of the resampling algorithm of [28], the swapping algorithm of [18], and all algorithms analyzed by entropy compression, is that they are instances of focused stochastic local search. The general idea in stochastic local search is that Ω is equipped with a neighborhood structure, so that the search for flawless objects starts at some (flawed) object (state) and moves stochastically from state to state along the neighborhood structure. Restricting the search so that every transition away from a state σ must target one of the flaws present in σ is known as *focusing* the search [30]. The first effort to give a convergence condition for focused stochastic local search algorithms with arbitrary transition probabilities, i.e., not mandated by a background measure μ, was the flaws/actions framework of [3]. As our work also uses this framework, below we recall some of the relevant definitions. Note that the existence of flawless objects is not presumed in any of these analyses. Instead, the idea is to establish the existence of flawless objects by proving that some focused stochastic local search algorithm (quickly) converges to one.

For $\sigma \in \Omega$, let $U(\sigma)$ denote the set of indices of the flaws present in σ, i.e., $U(\sigma) = \{i \in [m] : f_i \ni \sigma\}$. For every $i \in U(\sigma)$, let $A(i, \sigma) \neq \{\sigma\}$ be a nonempty subset

of Ω. The elements of $A(i,\sigma)$ are called *actions,* and we consider the multidigraph D on Ω that has an arc $\sigma \xrightarrow{i} \tau$ for every $\tau \in A(i,\sigma)$. We will consider walks on D which start at a state σ_1 selected according to some probability distribution θ and which, at each nonsink vertex σ, first select a flaw $f_i \ni \sigma$ as a function of the trajectory so far (focus), and then select as the next state $\tau \in A(i,\sigma)$ with probability $\rho_i(\sigma,\tau)$. Whenever flaw $f_i \ni \sigma$ is selected, we will say that flaw f_i was *addressed.* This will not necessarily mean that f_i will be eliminated, i.e., potentially $A(i,\sigma) \cap f_i \neq \emptyset$. The multidigraph D should be thought of as implicitly defined by the algorithm we wish to analyze each time, not as explicitly constructed. Also, when we refer to running "time," we mean the number of steps on D, without concern for exactly how long it takes to perform a single step, i.e., to identify a flaw present and select from its actions.

In this language, [3] gave a sufficient condition for algorithmic convergence when

(a) D is *atomic*; i.e., for every $\tau \in \Omega$ and every $i \in [m]$ there exists at most one arc $\sigma \xrightarrow{i} \tau$.

(b) ρ assigns *equal* probability to every action in $A(i,\sigma)$, for every $\sigma \in \Omega$ and $i \in U(\sigma)$.

By analyzing algorithms satisfying conditions (a) and (b), several results that had been proved by custom versions of the LLL, and thus fell outside the algorithmization framework of [28], were made constructive and improved in [3]. At the same time, the convergence condition of [3] makes it possible to recover most results of the entropic method by generic arguments (sometimes with a small parameter loss). Finally, it is worth pointing out that even though the framework of [3] does not reference a background probability measure μ, it captures a large fraction of the applications of general LLL. This is because when μ is uniform and bad events correspond to partial assignments, a very common scenario, the state transitions of the resampling algorithm of Moser and Tardos satisfy both conditions (a) and (b). Overall, however, the convergence condition of [3] was incomparable with those of the LLL algorithmizations preceding it.

The long line of work on LLL algorithmizations that started with the groundbreaking work of Moser culminated with the work of Harvey and Vondrák [19]. They showed that the lopsided LLL can be made constructive even under the most permissive condition (Shearer's) whenever we can construct efficient *resampling oracles.* Resampling oracles elegantly capture the common core of all LLL algorithmizations, namely that the state transitions, (D,ρ), are perfectly compatible with the background measure μ. Below we give the part of the definition of resampling oracles that exactly expresses this notion of compatibility, which we call (measure) *regeneration.*

REGENERATION (Harvey–Vondrák [19]). *Say that* (D,ρ) regenerate μ at flaw f_i *if for every* $\tau \in \Omega$,

$$\frac{1}{\mu(f_i)} \sum_{\sigma \in f_i} \mu(\sigma)\rho_i(\sigma,\tau) = \mu(\tau) \ . \tag{2}$$

Observe that the left-hand side (l.h.s.) of (2) is the probability of reaching state τ after first sampling a state $\sigma \in f_i$ according to μ and then addressing f_i at σ. The requirement that this probability equals $\mu(\tau)$ for every $\tau \in \Omega$ means that (D,ρ) must be such that in every state $\sigma \in f_i$ the distribution on actions for addressing f_i *perfectly* removes the conditional $f_i \ni \sigma$. Of course, a trivial way to satisfy this requirement is to sample a new state σ' according to μ in each step (assuming μ is efficiently

sampleable). Doing this, however, foregoes any notion of iterative progress towards a goal, as the set of flaws present in σ' is completely unrelated to those in σ. Instead, we would like to respect (2) while limiting the flaws introduced in σ'. Towards this end, we can consider the projection of the action digraph D capturing those flaws which may be introduced (caused) when we address each flaw. It is important to note that, below, potential causality is independent of flaw choice and that the causality digraph has an arc $i \to j$ if there exists *even one* transition aimed at addressing f_i that causes f_j to appear in the new state. Naturally, the sparser this causality digraph, the better.

POTENTIAL CAUSALITY. *For an arc $\sigma \xrightarrow{i} \tau$ in D and a flaw f_j present in τ, we say that f_i causes f_j if $f_i = f_j$ or $f_j \not\ni \sigma$. We say that f_i* potentially causes *f_j if D contains* at least one *arc wherein f_i causes f_j.*

CAUSALITY DIGRAPH. *The digraph $C = C(\Omega, F, D)$ on $[m]$ where $i \to j$ iff f_i potentially causes f_j is called the causality digraph. The* neighborhood *of a flaw f_i is $\Gamma(i) = \{j : i \to j$ exists in $C\}$.*

Harvey and Vondrák [19] proved that for essentially every lopsidependency digraph L of interest, there exist resampling oracles whose causality digraph is (a subgraph of) L. We should emphasize, however, that there is no guarantee that these promised resampling oracles can be implemented efficiently to yield an LLL algorithmization (and, naturally, in the absence of efficiency considerations, the LLL is already "algorithmic" by exhaustive search). Indeed, as we discuss below, there are settings in which the existence of efficient resampling oracles seems unlikely. That said, in [19] Harvey and Vondrák demonstrated the existence of efficient resampling oracles for a plethora of LLL applications in the variable setting, the permutation setting, and several other settings.

Perhaps the simplest demonstration of the restrictiveness of resampling oracles comes from one of the oldest and most vexing concerns about the LLL (see the survey of Szegedy [35]), namely, the inability[2] of the LLL to establish that a graph with maximum degree Δ can be colored with $q = \Delta + 1$ colors. For example, if μ is the uniform measure on all q^n colorings with q colors, then every time a vertex v is recolored, its color must be chosen uniformly from among all colors, something that induces a requirement of $q > \mathrm{e}\Delta$ colors. If, instead, we were to choose only from among colors that do not currently appear in v's neighborhood, then for all $q \geq \Delta+1$, the causality digraph would be empty, and rapid termination follows trivially. But it seems very hard to describe a probability measure μ and resampling oracles for it that respect the empty causality graph.

To recap, there are two schools of thought. In the first, we start from the central object of the LLL, the measure μ on Ω, and try to design an algorithm that moves from one state to another in a manner that perfectly respects the measure. In the second, there is no measure on Ω at all, and both the transitions and their probabilities can be, a priori, arbitrary. In this work, we bring these two schools of thought together by introducing the notion of *measure distortion*, showing, in particular, that the first corresponds to the special case of no distortion, and that the second corresponds

[2]Naturally, whenever the set of flawless objects Ω^* is nonempty, the uniform measure on Ω^* demonstrates the existence of flawless objects. So, in a trivial sense, there is nothing that cannot be established by the LLL. But, of course, anyone in possession of a description of Ω^* allowing the construction of a measure on it does not need the LLL. Indeed, the whole point of the LLL is that it offers incredibly rich conclusions, e.g., $\Omega^* \neq \emptyset$, from extremely meager ingredients, e.g., the uniform measure on Ω.

to the uniform measure but with distortion. The main point of our work, however, is to demonstrate that the generality afforded by allowing measure distortion has tangible benefits. Specifically, in complex applications, removing the conditioning that corresponds to the resampled bad event in each step may be impossible to do perfectly (as is required to have a resampling oracle) if one is only allowed short travel within Ω, i.e., "local search." This is because small, but nonvanishing, correlations can travel arbitrarily far in the structure. Allowing measure distortion removes the requirement of perfect deconditioning by accounting for such correlations through distortion, enabling a local analysis. This makes it possible to design natural, local algorithms and to prove rigorous mathematical statements about their convergence in the presence of long-range correlations.

Concretely, we extend the flaws/actions framework of [3] to allow arbitrary action digraphs D, arbitrary transition probabilities ρ, and incorporation of arbitrary background measures μ, allowing us to connect the flaws/actions framework to the LLL. Our work highlights the role of the measure μ in gauging how efficiently the algorithm rids the state of flaws, i.e., as a *gauge of progress*, by pointing out the trade-off between distortion and the sparsity of the causality graph. The end result is a theorem that subsumes both the results of [3] and the algorithmization of the lopsided LLL [19] via resampling oracles, establishing a uniform method for designing and analyzing focused stochastic local search algorithms. Additionally, our work makes progress on elucidating the role of flaw choice in stochastic local search, and establishes several structural facts about resampling oracles.

2. Statement of results. We develop tools for analyzing *focused stochastic local search* algorithms. Specifically, we establish a sequence of increasingly general conditions under which such algorithms find flawless objects quickly, presented as Theorems 1, 2, and 3. For the important special case of atomic action digraphs, we identify structural properties of resampling oracles, presented as Theorem 4. For the same setting, we also derive a sharp analysis for the probability of any trajectory, elucidating the role of flaw choice, presented as Theorem 5.

Theorems 1–3 differ in the sophistication of the flaw-choice mechanism they can accommodate. While in other works, such as [28] on the variable setting and [18] on permutations, the setting was sufficiently symmetric that flaw choice could be arbitrary, in more complex applications more sophisticated flaw choice is necessary. For example, to establish our results on acyclic edge coloring (AEC), we must use our recursive algorithm (Theorem 2), as the simple Markov walk (Theorem 1), let alone arbitrary flaw choice, will not work.

To demonstrate the flexibility of our framework, we derive a bound for AEC of graphs with bounded degeneracy, a class including all graphs of bounded treewidth, presented as Theorem 6 in section 6.1. To derive the result, we rely heavily on the actions *not* forming resampling oracles with respect to the measure used. Unlike other recent algorithmic work on the problem [13, 15], our result is established not with ideas/computations "customized" to the problem but rather as a direct application of Theorem 2, highlighting its capacity to incorporate both global conditions, such as degeneracy and sophisticated flaw-choice mechanisms, which in this case is a recursive procedure. We also show how to effortlessly derive an upper bound of $4.182(\Delta-1)$ for AEC of general graphs, which comes close to the hard-won bound of $4(\Delta-1)$ of Esperet and Parreau [13] via a custom analysis. Finally, we note that Iliopoulos [20] recently showed how our main theorem can be used to analyze the algorithm of Molloy [25] for coloring triangle-graph graphs of degree up to the "shattering threshold" for random graphs [1].

2.1. Setup. Recall that we consider algorithms which at each flawed state σ select some flaw $f_i \ni \sigma$ to address and then select the next state $\tau \in A(i,\sigma)$ with probability $\rho_i(\sigma,\tau)$. As one may expect, the flaw-choice mechanism does have a bearing on the running time of such algorithms, and we discuss this point in section 2.6. Our results focus on conditions for rapid termination that do not require sophisticated flaw choice (but can be used in conjunction with such choice).

To measure a walk's capacity to rid the state of flaws, we introduce a measure μ on Ω, as in the LLL. Without loss of generality, and to avoid certain trivialities, we assume that $\mu(\sigma) > 0$ for all $\sigma \in \Omega$. The choice of μ is entirely ours and can be oblivious, e.g., $\mu(\cdot) = |\Omega|^{-1}$. While μ typically assigns only exponentially small probability to flawless objects, it will allow us to prove that the walk reaches a flawless object in polynomial time with high probability.

To do this, we define a "charge" $\gamma_i = \gamma_i(D,\theta,\rho,\mu)$ for each flaw $f_i \in F$ that captures the *compatibility* between the actions of the algorithm for addressing flaw f_i and the measure μ. Specifically, just as for regeneration, we consider the probability $\nu_i(\tau)$ of ending up in state τ after (i) sampling a state $\sigma \in f_i$ according to μ, and (ii) addressing f_i at σ. But instead of requiring that $\nu_i(\tau)$ *equals* $\mu(\tau)$, as in resampling oracles, we allow $\nu_i(\tau)$ to be free and simply measure

$$d_i = \max_{\tau\in\Omega} \frac{\nu_i(\tau)}{\mu(\tau)} \geq 1 \ , \tag{3}$$

i.e., the greatest inflation of a state probability incurred by addressing f_i (relative to its probability under μ, and averaged over the initiating state $\sigma \in f_i$ according to μ). The charge γ_i of flaw f_i is then defined as

$$\gamma_i := d_i \cdot \mu(f_i) \tag{4}$$

$$= \max_{\tau\in\Omega} \frac{1}{\mu(\tau)} \sum_{\sigma\in f_i} \mu(\sigma)\rho_i(\sigma,\tau) \ . \tag{5}$$

To gain some intuition for γ_i, we observe that if μ is uniform and D is atomic, then γ_i is simply the greatest transition probability ρ_i on any arc originating in f_i.

To state our results, we need a definition regarding the distribution θ of the starting state.

Definition 1. *The* span *of a probability distribution $\theta : \Omega \to [0,1]$, denoted by $\mathcal{S}(\theta)$, is the set of flaw indices that may be present in a state selected according to θ, i.e., $\mathcal{S}(\theta) = \bigcup_{\sigma\in\Omega:\theta(\sigma)>0} U(\sigma)$.*

Finally, we discuss why (1) is equivalent to the standard form of the general LLL condition, namely $\mu(A_i) \leq x_i \prod_{j\in D(i)}(1-x_j)$, where $x_i \in [0,1)$, for every $i \in [m]$. Since the case $x_i = 0$ is uninteresting, we may assume $x_i \in (0,1)$. Letting $\psi_i > 0$, substituting $x_i = \psi_i/(1+\psi_i) \in (0,1)$ in the standard condition, and simplifying, we get $\mu(A_i)\prod_{j\in\{i\}\cup D(i)}(1+\psi_j) \leq \psi_i$. Opening up the product, we get (1).

2.2. A simple Markov chain. Our first result concerns the simplest case, where in each flawed state σ, the algorithm addresses the greatest flaw present in σ, according to an arbitrary but fixed permutation of the flaws. Recall that μ is the measure on Ω used to measure progress, γ_i is the charge of flaw f_i according to μ, and θ is the starting state distribution.

THEOREM 1. *If there exist positive real numbers* $\{\psi_i\}_{i\in[m]}$ *such that for every* $i \in [m]$,

$$\zeta_i := \frac{\gamma_i}{\psi_i} \sum_{S\subseteq\Gamma(i)} \prod_{j\in S} \psi_j < 1 \ , \tag{6}$$

then for every permutation π, *the walk reaches a sink within* $(T_0 + s)/\delta$ *steps with probability at least* $1 - 2^{-s}$, *where* $\delta = 1 - \max_{i\in[m]} \zeta_i > 0$, *and*

$$T_0 = \log_2\left(\max_{\sigma\in\Omega} \frac{\theta(\sigma)}{\mu(\sigma)}\right) + \log_2\left(\sum_{S\subseteq\mathcal{S}(\theta)} \prod_{j\in S} \psi_j\right) = \log_2\left(\max_{\sigma\in\Omega} \frac{\theta(\sigma)}{\mu(\sigma)}\right) + \sum_{j\in\mathcal{S}(\theta)} \log_2(1+\psi_j) \ .$$

Theorem 1 has two features worth discussing that are shared by all our results.

Arbitrary starting state. Since θ can be arbitrary, any foothold on Ω suffices to apply the theorem. Note also that T_0 captures the trade-off between starting at a fixed state vs. starting at a state sampled from μ. In the latter case, i.e., when $\theta = \mu$, the first term in T_0 vanishes, but the second term grows to reflect the uncertainty of the set of flaws present in σ_1.

Arbitrary number of flaws. The running time depends only on the span $|\mathcal{S}(\theta)|$, not the total number of flaws $|F|$. This has an implication analogous to the result of Hauepler, Saha, and Srinivasan [17] on core events: even when $|F|$ is super-polynomial in the problem's encoding length, it may still be possible to get a polynomial-time algorithm. For example, this can be done by proving that in every state only polynomially many flaws may be present, or by finding a specific state σ_1 such that $|U(\sigma_1)|$ is small.

2.3. A non-Markovian algorithm. Our next results concerns the common setting, where the neighbors of each flaw in the causality graph span several arcs between them. We improve Theorem 1 in such a setting by employing a *recursive* algorithm, that is, an algorithm in which the flaw choice at each step depends on the *entire* trajectory up to that point, not just the current state, so that the resulting walk on Ω is 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. The improvement is that rather than summing over all subsets of $\Gamma(i)$ as in (6), we now only sum over *independent* such subsets, where f_i, f_j are dependent if $i \to j$ *and* $j \to i$. This improvement is similar to the cluster expansion improvement of Bissacot et al. [6] of the general LLL. As a matter of fact, Theorem 2 implies the algorithmic aspects of [6] (see [32, 19]).

Further, the use of a recursive algorithm makes it possible to "shift responsibility" between flaws, so that gains from the aforementioned restriction of the sum can be realized by purposeful flaw ordering. For a permutation π of F, let $I_\pi(S)$ denote the index of the greatest flaw in any $S \subseteq F$ according to π. For a fixed action digraph D with causality digraph C, the recursive algorithm *takes as input* any digraph $R \supseteq C$, i.e., any supergraph of C, and is the non-Markovian random walk on Ω that occurs by invoking procedure ELIMINATE. Observe that if in line 8 we do not intersect $U(\sigma)$ with $\Gamma_R(i)$, then the recursion is trivialized, recovering the simple walk of Theorem 1. Its convergence condition, Theorem 2, involves sums over the independent sets of R, generalizing the discussion above (as one can always take $R = C$).

The reason for allowing the addition of arcs in R relative to C is that while adding, say, arcs $i \to j$ and $j \to i$ may make the sums corresponding to f_i and f_j greater, if

Recursive Walk

```
1: procedure ELIMINATE
2:     σ ← θ(·)                                   ▷ Sample σ from θ
3:     while U(σ) ≠ ∅ do
4:         ADDRESS (I_π(U(σ)), σ)
5:     return σ
6: procedure ADDRESS(i, σ)
7:     σ ← τ ∈ A(i, σ) with probability ρ_i(σ, τ)
8:     while B = U(σ) ∩ Γ_R(i) ≠ ∅ do             ▷ Note ∩Γ_R(f_i)
9:         ADDRESS(I_π(B), σ)
```

flaw f_k is such that $\{i,j\} \subseteq \Gamma(k)$, then the sum for flaw f_k may become smaller, since f_i, f_j are now dependent. As a result, without modifying the algorithm, such an arc addition can help establish a sufficient condition for rapid convergence to a flawless object, e.g., in our application on AEC in section 6. An analogous phenomenon appears in the improvement of Bissacot et al. [6], i.e., denser dependency graphs may yield better analysis.

DEFINITION 2. *For a digraph R on $[m]$, let $G = G(R) = ([m], E)$ be the* undirected *graph, where $\{i,j\} \in E$ iff both $i \to j$ and $j \to i$ exist in R. For $S \subseteq F$, let* $\mathrm{Ind}(S) = \{S' \subseteq S : S' \text{ is an independent set in } G\}$.

THEOREM 2. *Let $R \supseteq C$ be arbitrary. If there exist positive real numbers $\{\psi_i\}$ such that for every $i \in [m]$,*

$$\zeta_i := \frac{\gamma_i}{\psi_i} \sum_{S \in \mathrm{Ind}(\Gamma_R(i))} \prod_{j \in S} \psi_j < 1 \ , \tag{7}$$

then for every permutation π, Recursive Walk reaches a sink within $(T_0 + s)/\delta$ steps with probability at least $1 - 2^{-s}$, where $\delta = 1 - \max_{i \in [m]} \zeta_i > 0$, and

$$T_0 = \log_2 \left(\max_{\sigma \in \Omega} \frac{\theta(\sigma)}{\mu(\sigma)} \right) + \log_2 \left(\sum_{S \subseteq \mathrm{Ind}(\mathcal{S}(\theta))} \prod_{j \in S} \psi_j \right) \ .$$

Remark 1. Theorem 2 strictly improves Theorem 1 because for $R = C$, (i) the summation in (7) is only over the subsets of $\Gamma_R(i)$ that are independent in G instead of all subsets of $\Gamma_R(i)$ as in (6); and (ii) similarly for T_0, the summation is only over the independent subsets of $\mathcal{S}(\theta)$ rather than all subsets of $\mathcal{S}(\theta)$.

Remark 2. Theorem 2 can be strengthened by introducing for each flaw $f_i \in F$ a permutation π_i of $\Gamma_R(i)$ and replacing π with π_i in line 9 of Recursive Walk. With this change in (7), it suffices to sum only over $S \subseteq \Gamma_R(i)$, satisfying the following: if the subgraph of R induced by S contains an arc $j \to k$, then $\pi_i(j) \geq \pi_i(k)$. As such a subgraph cannot contain both $j \to k$ and $k \to j$, we see that $S \in \mathrm{Ind}(\Gamma_R(i))$.

2.4. A general theorem. Theorems 1 and 2 are instantiations of a general theorem we develop for analyzing focused stochastic local search algorithms. Before stating the theorem, we briefly discuss its derivation in order to motivate its form. Recall that a focused local search algorithm $\mathcal{A}$ amounts to a flaw-choice mechanism driving a random walk on a multidigraph D with transition probabilities ρ and starting state distribution θ.

To bound the probability that $\mathcal{A}$ runs for t or more steps, we partition the set of all t-trajectories into equivalence classes, bound the total probability of each class, and sum the bounds for the different classes. Specifically, the partition is formed according to the t-sequence of the first t flaws addressed.

DEFINITION 3. *For any integer $t \geq 1$, let $\mathcal{W}_t(\mathcal{A})$ denote the set containing all t-sequences of flaws that have positive probability of being the first t flaws addressed by $\mathcal{A}$.*

In general, the content of $\mathcal{W}_t(\mathcal{A})$ is an extremely complex function of flaw choice. An essential idea of our analysis is to estimate it by *syntactic* considerations capturing the following necessary condition for $W \in \mathcal{W}_t(\mathcal{A})$: while the very first occurrence of any flaw f_j in W may be attributed to $f_j \ni \sigma_1$, every subsequent occurrence of f_j must be preceded by a distinct occurrence of a flaw f_i that "assumes responsibility" for f_j, e.g., a flaw f_i that potentially causes f_j. Definition 4 below establishes a framework for bounding $\mathcal{W}_t(\mathcal{A})$ by relating flaw choice with responsibility (i) by requiring that the flaw-choice mechanism is such that the elements of $\mathcal{W}_t(\mathcal{A})$ can be unambiguously represented as forests with t vertices, while on the other hand, (ii) by generalizing the subsets of flaws for which a flaw f_i may be responsible from subsets of $\Gamma(i)$ to arbitrary subsets of flaws, thus enabling responsibility shifting.

DEFINITION 4. *We will say that algorithm $\mathcal{A}$ is* traceable *if there exist sets* $\mathrm{Roots}(\theta) \subseteq 2^{[m]}$ *and* $\mathrm{List}(1) \subseteq 2^{[m]}, \ldots, \mathrm{List}(m) \subseteq 2^{[m]}$ *such that for every $t \geq 1$, the flaw sequences in $\mathcal{W}_t(\mathcal{A})$ can be injected into unordered rooted forests with t vertices that have the following properties:*

1. *Each vertex of the forest is labeled by an integer $i \in [m]$.*
2. *The labels of the roots of the forest are distinct and form an element of* $\mathrm{Roots}(\theta)$.
3. *The indices labeling the children of each vertex are distinct.*
4. *If a vertex is labeled by $i \in [m]$, then the labels of its children form an element of* $\mathrm{List}(i)$.

In [3] it was shown that both the simple random walk algorithm in Theorem 1 and Recursive Walk in Theorem 2 are traceable. Specifically, the set $\mathcal{W}_t$ of the former can be injected into *break forests*, so that Definition 4 is satisfied, with $\mathrm{Roots}(\theta) = 2^{\mathcal{S}(\theta)}$ and $\mathrm{List}(i) = 2^{\Gamma_R(i)}$. For the latter, $\mathcal{W}_t$ can be analogously injected into *recursive forests* with $\mathrm{Roots}(\theta) = \mathrm{Ind}(\mathcal{S}(\theta))$ and $\mathrm{List}(i) = \mathrm{Ind}(\Gamma_R(i))$. Thus, Theorems 1 and 2 follow readily from Theorem 3 below.

THEOREM 3 (main result). *If algorithm $\mathcal{A}$ is traceable and there exist positive real numbers $\{\psi_i\}_{i\in[m]}$ such that for every $i \in [m]$,*

$$\zeta_i := \frac{\gamma_i}{\psi_i} \sum_{S \in \mathrm{List}(i)} \prod_{j \in S} \psi_j < 1 \ , \tag{8}$$

then $\mathcal{A}$ reaches a sink within $(T_0 + s)/\delta$ steps with probability at least $1 - 2^{-s}$, where $\delta = 1 - \max_{i\in[m]} \zeta_i$ and

$$T_0 = \log_2 \left(\max_{\sigma \in \Omega} \frac{\theta(\sigma)}{\mu(\sigma)} \right) + \log_2 \left(\sum_{S \in \mathrm{Roots}(\theta)} \prod_{j \in S} \psi_j \right) \ .$$

Theorem 3 also implies the "left-handed random walk" result of [3] and extends it to nonuniform transition probabilities, since that algorithm is also traceable. Notably,

in the left-handed LLL introduced by Pedgen [31] and which inspired the algorithm, the flaw order π can be chosen in a *provably* beneficial way, unlike in the algorithms of Theorems 1 and 2, which are indifferent to π. Establishing this goodness, however, entails attributing responsibility very differently from what is suggested by the causality digraph, making full use of the power afforded by traceability and Theorem 3.

2.5. Resampling oracles via atomic actions. To get a constructive result by LLL algorithmization via resampling oracles, i.e., given Ω, F, and μ, we must design (D, ρ) that regenerate μ at every flaw $f_i \in F$. This can be a daunting task in general. We simplify this task greatly for atomic action digraphs. Such digraphs capture algorithms that appear in several settings, e.g., the Moser–Tardos algorithm when flaws correspond to partial assignments, the algorithm of Harris and Srinivasan for permutations [18], and others (see [3]). While atomicity may seem like an artificial condition, it is actually a natural way to promote search space exploration, as it is equivalent to the following: distinct states $\sigma, \sigma' \in f_i$ must have disjoint actions, i.e., $A(i,\sigma) \cap A(i,\sigma') = \emptyset$. In most settings, atomicity can be achieved in a straightforward manner. For example, in the variable setting, atomicity is implied by an idea that is extremely successful in practice, namely "focus" [30, 33, 4]: every state transformation should be the result of selecting a flaw present in the current state and modifying only the variables of that flaw.

Theorem 4 asserts that when the action digraph D must be atomic, then in order to regenerate μ at f_i it is sufficient (and necessary) for the states in each set $A(i,\sigma)$ to have total probability given by (9). Equation (10) then automatically provides appropriate transition probabilities. Combined, (9) and (10) offer strong guidance in designing resampling oracles in atomic digraphs.

THEOREM 4. *If D is atomic and (D, ρ) regenerate μ at f_i, then for every $\sigma \in f_i$,*

$$\sum_{\tau \in A(i,\sigma)} \mu(\tau) = \frac{\mu(\sigma)}{\mu(f_i)}, \tag{9}$$

$$\rho_i(\sigma,\tau) = \frac{\mu(\tau)}{\sum_{\sigma' \in A(i,\sigma)} \mu(\sigma')} \quad \text{for every } \tau \in A(i,\sigma)\ . \tag{10}$$

2.6. A sharp analysis and the role of flaw choice. Let W_t be the random variable that equals the sequence of the first t flaws addressed by the algorithm, or $\perp$ if the algorithm reaches a flawless object in fewer than t steps. Recall that $\mathcal{W}_t(\mathcal{A})$ denotes the set of all t-sequences of flaws that have positive probability of being the first t flaws addressed by an algorithm $\mathcal{A}$, i.e., the range of W_t except $\perp$. Trivially, the probability that $\mathcal{A}$ takes at least t steps equals

$$\sum_{W \in \mathcal{W}_t(\mathcal{A})} \Pr[W_t = W]\ .$$

THEOREM 5. *For any algorithm $\mathcal{A}$ for which D is atomic and (D, ρ) regenerate μ at every flaw, for every flaw sequence $W = w_1, w_2, \ldots, w_t \in \mathcal{W}_t(\mathcal{A})$,*

$$\Pr[W_t = W] \in [\alpha, \beta] \cdot \prod_{i=1}^{t} \mu(w_i)\ , \tag{11}$$

where $\alpha = \min_{\sigma \in \Omega} \theta(\sigma)/\mu(\sigma)$ and $\beta = \max_{\sigma \in \Omega} \theta(\sigma)/\mu(\sigma)$.

Theorem 5 tells us that every algorithm in which (D, ρ) form atomic resampling oracles will converge to a flawless object *if and only if* the sum

$$\sum_{W \in \mathcal{W}_t(\mathcal{A})} \prod_{i=1}^{t} \mu(w_i)$$

converges to zero as t grows. In other words, the quality of the algorithm depends *solely* on the set $\mathcal{W}_t(\mathcal{A})$ which, in turn, is determined by flaw choice (and the initial distribution θ).

In the work of Moser and Tardos for the variable setting [28] and of Harris and Srinivasan for the uniform measure on permutations [18], flaw choice can be arbitrary, and the whole issue "is swept under the rug" [35]. This can be explained as follows. In these settings, due to the symmetry of Ω, we can afford to bound $\mathcal{W}_t(\mathcal{A})$ in a way that completely ignores flaw choice; i.e., we can deem flaw choice to be adversarial but still recover the LLL condition. Theorem 5 shows that this should not be confused with deeming flaw choice "irrelevant" for such algorithms. In fact, exactly the opposite is true, and this was also established in [8], which showed that the Moser–Tardos algorithm, in practice, succeeds on instances far denser than predicted by the LLL condition.

Kolmogorov [23] gave a sufficient condition, called *commutativity*, for arbitrary flaw choice. One can think of commutativity as the requirement that there exist a supergraph of the causality graph satisfying a strong symmetry condition (including that all arcs be bidirectional), for which the LLL condition still holds. However, such symmetries cannot be expected to hold in general, which is reflected in the requirement of traceability in our Theorem 3 and in the specificity of the flaw-choice mechanisms in our Theorems 1 and 2. More generally, in [19], Harvey and Vondrák provided strong evidence that in the absence of commutativity, a specific flaw choice is necessary to match Shearer's criterion for the LLL.

2.7. Comparison with resampling oracles. Harvey and Vondrák [19] proved that in the setting of resampling oracles, i.e., no distortion, when the causality graph is symmetric, if one resamples a maximal independent set of bad events each time, the resulting algorithm succeeds even under Shearer's condition. (Notably, Shearer's condition, involving an exponential number of terms, is not used in applications.) As a corollary, they prove that in this setting, in (6), strict inequality ($<$) can be replaced with inequality ($\leq$). As our results are over arbitrary directed causality graphs, for which no analogue to Shearer's condition exists, we do not have an analogous result. However, for the case when the causality graph is symmetric (undirected), Kolmogorov showed [23] that the analogue of Shearer's lemma holds in our framework. That is, if the conditions that result when in the standard Shearer lemma one replaces probabilities with charges are satisfied, then the algorithm that resamples a maximal independent set of bad events each time succeeds.

As a final remark, we note that there are settings in which the existence of resampling oracles seems unlikely mainly for two reasons. First, there are cases when it is known that resampling oracles do exist but cannot be made efficient, because of an existing hardness result for the main search problem (see [19] for details and examples). In these cases, there is really nothing we can do. (Note also that all our convergence results bound the number of steps of the walk in D and do not consider the running time to perform each step.) Second, it could be the case that resampling oracles do not exist because the related LLL condition is not satisfied, but some relaxation of this condition is satisfied, which allows one to design an oracle preserving

the outgoing measure approximately. This is exactly the situation which our result handles better than [19].

3. Bounding the probabilities of trajectories. To bound the probability that an algorithm $\mathcal{A}$ runs for t or more steps we, partition its t-trajectories into equivalence classes, bound the total probability of each class, and sum the bounds for the different classes. Formally, for a trajectory $\Sigma = \sigma_1 \xrightarrow{w_1} \sigma_2 \xrightarrow{w_2} \cdots$, we let $W(\Sigma) = w_1, w_2, \ldots$ denote its *witness* sequence, i.e., the sequence of flaws addressed along Σ (note that Σ determines $W(\Sigma)$, as the flaw choice is deterministic). We let $W_t(\Sigma) = \perp$ if Σ has fewer than t steps; otherwise, we let $W_t(\Sigma)$ be the t-prefix of $W(\Sigma)$. Slightly abusing notation, as mentioned, we let $W_t = W_t(\Sigma)$ be the random variable when Σ is the trajectory of the walk, i.e., selected according to (D, ρ, θ) and the flaw-choice mechanism. Finally, recall that $\mathcal{W}_t = \mathcal{W}_t(\mathcal{A})$ denotes the range of W_t for algorithm $\mathcal{A}$ except for $\perp$; i.e., $\mathcal{W}_t(\mathcal{A})$ is the set of t-sequences of flaws that have positive probability of being the first t flaws addressed by $\mathcal{A}$ per Definition 3. Thus,

$$\Pr[\text{Algorithm } \mathcal{A} \text{ takes } t \text{ or more steps}] = \sum_{W \in \mathcal{W}_t(\mathcal{A})} \Pr[W_t = W] \ .$$

Key to our analysis will be the derivation of an upper bound for $\Pr[W_t = W]$ that holds for *arbitrary* t-sequences of flaws, i.e., not necessarily elements of $\mathcal{W}_t(\mathcal{A})$, and which factorizes over the flaws in W. For an arbitrary sequence of flaws $A = a_1, \ldots, a_t$, let us denote by $[i]$ the index $j \in [m]$ such that $a_i = f_j$.

LEMMA 1. *Let $\xi = \xi(\theta, \mu) = \max_{\sigma \in \Omega}\{\theta(\sigma)/\mu(\sigma)\}$. For every sequence of flaws $W = w_1, \ldots, w_t$,*

$$\Pr[W_t = W] \leq \xi \prod_{i=1}^{t} \gamma_{[i]} \ .$$

Proof. We claim that for every $t \geq 0$, every t-sequence of flaws W, and every state $\tau \in \Omega$,

$$\Pr[W_t = W \cap \sigma_{t+1} = \tau] \leq \xi \cdot \prod_{i=1}^{t} \gamma_{[i]} \cdot \mu(\tau) \ . \tag{12}$$

Summing (12) over all $\tau \in \Omega$ proves the lemma.

To prove our claim, we proceed by induction on $|W|$ after recalling that for every $i \in [m]$ and $\tau \in \Omega$, by the definition of $\{\gamma_i\}_{i \in [m]}$,

$$\sum_{\sigma \in f_i} \mu(\sigma)\rho_i(\sigma, \tau) \leq \gamma_i \cdot \mu(\tau) \ . \tag{13}$$

For $|W| = 0$, the claim holds because $\Pr[\sigma_1 = \tau] = \theta(\tau) \leq \xi\mu(\tau)$ for all $\tau \in \Omega$, by the definition of ξ.

Assume that (12) holds for all s-sequences of flaws for some $s \geq 0$. Let $A' = A, f_i$ be any sequence of $s+1$ flaws, and let $\tau \in \Omega$ be arbitrary. The first inequality below is due to the fact that since f_i is the last flaw in A', a necessary (but not sufficient) condition for the event $W_{s+1} = A'$ to occur is that f_i is present in the state that results after the flaws in A have been addressed (it is not sufficient, as $\mathcal{A}$ may choose to address a flaw other than f_i). The second inequality follows from the inductive

hypothesis, while the third from (13).

$$\begin{aligned}\Pr[W_{s+1} = A' \cap \sigma_{s+2} = \tau] &\leq \sum_{\sigma \in f_i} \rho_i(\sigma,\tau) \Pr[W_s = A \cap \sigma_{s+1} = \sigma] \\ &\leq \xi \cdot \prod_{i=1}^{s} \gamma_{[i]} \cdot \sum_{\sigma \in f_i} \mu(\sigma) \cdot \rho_i(\sigma,\tau) \\ &\leq \xi \cdot \prod_{i=1}^{s+1} \gamma_{[i]} \cdot \mu(\tau) \ . \end{aligned}$$

∎

4. Proofs of Theorems 4 and 5. First, for every digraph–measure pair (D,μ), we identify certain transition probabilities ρ as special.

HARMONIC WALKS. (D,ρ,μ) *are* harmonic *if for every* $i \in [m]$ *and every transition* $(\sigma,\tau) \in f_i \times A(i,\sigma)$,

$$\rho_i(\sigma,\tau) = \frac{\mu(\tau)}{\sum_{\sigma' \in A(i,\sigma)} \mu(\sigma')} \propto \mu(\tau) \ . \tag{14}$$

In words, when (D,ρ,μ) are harmonic, ρ_i assigns to each state in $A(i,\sigma)$ probability proportional to its probability under μ. It is easy to see that (D,ρ,μ) are harmonic in both the celebrated algorithm of Moser and Tardos [28] for the variable setting and in the algorithm of Harris and Srinivasan [18] for the uniform measure on permutations. What makes harmonic (D,ρ,μ) combinations special is that for any pair (D,μ), taking ρ so that (D,ρ,μ) are harmonic can be easily seen to minimize the expression

$$\max_{\tau \in A(i,\sigma)} \left\{ \rho_i(\sigma,\tau) \frac{\mu(\sigma)}{\mu(\tau)} \right\}$$

for every $\sigma \in f_i$ *simultaneously*. For atomic D, this suffices to minimize the charge γ_i over all possible ρ.

Proof of Theorem 4. If D is atomic, $\mu > 0$, and (D,ρ) regenerate μ at every flaw f_i, it follows that for every $\tau \in \Omega$, there is *exactly* one $\sigma \in f_i$ such that $\rho_i(\sigma,\tau) > 0$ (and also that $\bigcup_{\sigma \in f_i} A(i,\sigma) = \Omega$). Therefore, regeneration at f_i in this setting is equivalent to the following:

(15)

$$\text{For every } \tau \in \Omega \text{ and the unique } \sigma \text{ such that } \tau \in A(i,\sigma), \quad \rho_i(\sigma,\tau) = \mu(\tau)\frac{\mu(f_i)}{\mu(\sigma)} \ .$$

(Note that for given D,μ, there may be no ρ satisfying (15), as we also need that $\sum_{\tau \in A(i,\sigma)} \rho_i(\sigma,\tau) = 1$.)

Since $\rho_i(\sigma,\tau) \propto \mu(\tau)$ in (15), we get (10). Summing (15) over $\tau \in A(i,\sigma)$ yields (9). ∎

Proof of Theorem 5. Lemma 1, valid for any (D,ρ,μ,θ), readily yields the upper bound. For the lower bound, we start by recalling that in the proof of Theorem 4, we showed that if D is atomic and (D,ρ) regenerate μ at f_i, then $\bigcup_{\sigma \in f_i} A(i,\sigma) = \Omega$. Therefore, if $W \in \mathcal{W}_t$, since (D,ρ) regenerate μ at every f_i, for every $\tau \in \Omega$ there exists $\Sigma^\tau = \sigma_1^\tau, \ldots, \sigma_{t+1}^\tau$ such that $W(\Sigma^\tau) = W$ and $\sigma_{t+1}^\tau = \tau$. Trivially,

$$\Pr[\Sigma^\tau] = \theta(\sigma_1^\tau) \prod_{i=1}^{t} \rho_{[i]}(\sigma_i^\tau, \sigma_{i+1}^\tau) \ .$$

Since D is atomic and (D, ρ) regenerate μ at every flaw, (15) applies, yielding

$$\rho_{[i]}(\sigma_i^\tau, \sigma_{i+1}^\tau) = \mu(w_i)\frac{\mu(\sigma_{i+1}^\tau)}{\mu(\sigma_i^\tau)} \ .$$

Thus, by telescoping, we get

$$\Pr[\Sigma^\tau] = \theta(\sigma_1^\tau)\prod_{i=1}^{t}\mu(w_i)\frac{\mu(\sigma_{i+1}^\tau)}{\mu(\sigma_i^\tau)} = \frac{\theta(\sigma_1^\tau)}{\mu(\sigma_1^\tau)}\mu(\tau)\prod_{i=1}^{t}\mu(w_i) \ . \tag{16}$$

Summing (16) over $\tau \in \Omega$ gives the lower bound

$$\Pr[W_t = W] \geq \min_{\sigma\in\Omega}\frac{\theta(\sigma)}{\mu(\sigma)}\prod_{i=1}^{t}\mu(w_i) \ . \qquad \square$$

5. Proof of Theorem 3. Per the hypothesis of Theorem 3, the sequences in $\mathcal{W}_t$ can be injected into a set of rooted forests with t vertices that satisfy the properties of Definition 4. Let $\widetilde{\mathcal{W}}_t \supseteq \mathcal{W}_t$ be the set of *all* forests with t vertices that satisfy the properties of Definition 4. By Lemma 1, to prove the theorem it suffices to prove that $\max_{\sigma\in\Omega}\frac{\theta(\sigma)}{\mu(\sigma)}\sum_{W\in\widetilde{\mathcal{W}}_t}\prod_{i=1}^{t}\gamma_{[i]}$ is exponentially small in s for $t = T_0 + s$.

To proceed, we use ideas from [32]. Specifically, we introduce a branching process that produces only forests in $\widetilde{\mathcal{W}}_t$ and bound $\sum_{W\in\widetilde{\mathcal{W}}_t}\prod_{i=1}^{t}\gamma_{[i]}$ by analyzing it. Given any real numbers $0 < \psi_i < \infty$, we define $x_i = \frac{\psi_i}{\psi_i+1}$ and write $\mathrm{Roots}(\theta) = \mathrm{Roots}$ to simplify notation. Recall that neither the trees in each forest nor the nodes inside each tree are ordered. To start the process, we produce the roots of the labeled forest by rejection sampling as follows: For each $i \in [m]$ independently, with probability x_i we add a root with label i. If the resulting set of roots is in Roots, we accept the birth. If not, we delete the roots created and try again. In each subsequent round, we follow a very similar procedure. Specifically, at each step, each node u with label ℓ "gives birth," again by rejection sampling as follows: For each integer $i \in [m]$, independently, with probability x_i we add a vertex with label i as a child of u. If the resulting set of children of u is in $\mathrm{List}(\ell)$, we accept the birth. If not, we delete the children created and try again. It is not hard to see that this process creates every forest in $\widetilde{\mathcal{W}}_t$ with positive probability. Specifically, for a vertex labeled by ℓ, every set $S \notin \mathrm{List}(\ell)$ receives probability 0, while every set $S \in \mathrm{List}(\ell)$ receives probability proportional to

$$w(S) = \prod_{g\in S} x_g \prod_{h\in[m]\setminus S}(1 - x_h) \ .$$

To express the exact probability received by each $S \in \mathrm{List}(\ell)$, we define

$$Q(S) := \frac{\prod_{g\in S} x_g}{\prod_{g\in S}(1 - x_g)} = \prod_{g\in S}\psi_g \tag{17}$$

and let $Z = \prod_{i\in[m]}(1 - x_i)$. We claim that $w(S) = Q(S)\,Z$. To see the claim, observe that

$$\frac{w(S)}{Z} = \frac{\prod_{g\in S} x_g \prod_{h\in[m]\setminus S}(1 - x_h)}{\prod_{i\in[m]}(1 - x_i)} = \frac{\prod_{g\in S} x_g}{\prod_{g\in S}(1 - x_g)} = Q(S) \ .$$

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

$$\frac{w(S)}{\sum_{B\in\mathrm{List}(\ell)} w(B)} = \frac{Q(S)Z}{\sum_{B\in\mathrm{List}(\ell)} Q(B)Z} = \frac{Q(S)}{\sum_{B\in\mathrm{List}(\ell)} Q(B)} \ . \tag{18}$$

Similarly, each set $R \in \mathrm{Roots}$ receives probability equal to $Q(R)\big(\sum_{B\in\mathrm{Roots}} Q(B)\big)^{-1}$.

For each forest $\phi \in \widetilde{\mathcal{W}}_t$ and each node v of ϕ, let $N(v)$ denote the set of labels of its children, and let $\mathrm{List}(v) = \mathrm{List}(\ell)$, where ℓ is the label of v.

LEMMA 2. *The branching process described above produces every forest $\phi \in \widetilde{\mathcal{W}}_t$ with probability*

$$p_\phi = \left(\sum_{S\in\mathrm{Roots}} \prod_{i\in S} \psi_i\right)^{-1} \prod_{v\in\phi} \frac{\psi_v}{\sum_{S\in\mathrm{List}(v)} Q(S)}.$$

Proof. Let R denote the roots of ϕ. By (18),

$$\begin{aligned} p_\phi &= \frac{Q(R)}{\sum_{S\in\mathrm{Roots}} Q(S)} \prod_{v\in\phi} \frac{Q(N(v))}{\sum_{S\in\mathrm{List}(v)} Q(S)} \\ &= \frac{Q(R)}{\sum_{S\in\mathrm{Roots}} Q(S)} \cdot \frac{\prod_{v\in\phi\setminus R} \psi_v}{\prod_{v\in\phi} \sum_{S\in\mathrm{List}(v)} Q(S)} \\ &= \left(\sum_{S\in\mathrm{Roots}} Q(S)\right)^{-1} \prod_{v\in\phi} \frac{\psi_v}{\sum_{S\in\mathrm{List}(v)} Q(S)} \ . \end{aligned}$$

∎

Now note that

$$\begin{aligned} \sum_{W\in\widetilde{\mathcal{W}}_t} \prod_{i=1}^{t} \gamma_{[i]} &= \sum_{W\in\widetilde{\mathcal{W}}_t} \prod_{i=1}^{t} \frac{\zeta_{[i]}\,\psi_{[i]}}{\sum_{S\in\mathrm{List}([i])} Q(S)} \\ &\le \left(\max_{i\in[m]} \zeta_i\right)^t \sum_{W\in\widetilde{\mathcal{W}}_t} \prod_{i=1}^{t} \frac{\psi_{[i]}}{\sum_{S\in\mathrm{List}([i])} Q(S)} \\ &= \left(\max_{i\in[m]} \zeta_i\right)^t \sum_{W\in\widetilde{\mathcal{W}}_t} \left(p_W \sum_{S\in\mathrm{Roots}} Q(S)\right) \\ &= \left(\max_{i\in[m]} \zeta_i\right)^t \sum_{S\in\mathrm{Roots}} Q(S) \ . \end{aligned} \tag{19}$$

Using (19), we see that the binary logarithm of the probability that the walk does not encounter a flawless state within t steps is at most $t \log_2 (\max_{i\in F} \zeta_i) + T_0$, where

$$T_0 = \log_2 \left(\max_{\sigma\in\Omega} \frac{\theta(\sigma)}{\mu(\sigma)}\right) + \log_2 \left(\sum_{S\in\mathrm{Roots}} \prod_{i\in S} \psi_i\right) \ .$$

Therefore, if $t = (T_0 + s)/\log_2(1/\max_{i\in F} \zeta_i) \le (T_0 + s)/\delta$, the probability that the random walk on D does not reach a flawless state within t steps is at most 2^{-s}.

6. Application to acyclic edge coloring.

6.1. Earlier works and statement of result. An edge coloring of a graph is *proper* if all edges incident to each vertex have distinct colors. A proper edge coloring is *acyclic* if it has no bichromatic cycles, i.e., no cycle receives exactly two (alternating) colors. AEC was originally motivated by the work of Coleman and colleagues [10, 9] on the efficient computation of Hessians. The smallest number of colors, $\chi'_a(G)$, for which a graph G has an AEC can also be used to bound other parameters, such as the oriented chromatic number [24] and the star chromatic number [14], both of which have many practical applications. The first general linear upper bound for χ'_a was given by Alon [5], who proved $\chi'_a(G) \leq 64\Delta(G)$, where $\Delta(G)$ denotes the maximum degree of G. This bound was improved to 16Δ by Molloy and Reed [26], and then to $9.62(\Delta - 1)$ by Ndreca, Procacci, and Scoppola [29]. Interest in the problem was recently renewed due to the work of Esperet and Parreau [13], who proved $\chi'_a(G) \leq 4(\Delta - 1)$ via an entropy compression argument, a technique that goes beyond what the LLL can give for the problem. Very recently, Giotis et al. [15] improved the result of [13] to 3.74Δ.

We give a bound of $2\Delta + O(\sqrt{d/\Delta})$ for (simple) graphs of degeneracy bounded by d. Recall that a graph G is d-degenerate if its vertices can be ordered so that every vertex has at most d neighbors greater than itself. Thus, we not only cover a significant class of graphs but also demonstrate that our method can incorporate global graph properties. For example, if $\mathcal{G}_d$ denotes the set of all d-degenerate graphs, then all planar graphs are in $\mathcal{G}_5$, while all graphs with treewidth or pathwidth at most d are in $\mathcal{G}_d$ (for more on degenerate graphs, see [21]). We prove the following.

THEOREM 6. *Every d-degenerate graph of maximum degree Δ has an AEC with $\lceil (2+\epsilon)\Delta \rceil$ colors that can be found in polynomial time, where $\epsilon = 4\sqrt{d/\Delta}$.*

6.2. Background. As will become clear shortly, the main difficulty in AEC comes from the short cycles of G, with 4-cycles being the toughest. This motivates the following definition.

DEFINITION 5. *Given a graph $G = (V, E)$ and a, perhaps partial, edge coloring of G, say that color c is* 4-forbidden *for $e \in E$ if assigning c to e would result in either a violation of proper edge coloring, or a bichromatic 4-cycle containing e. Say that c is 4-available if it is not 4-forbidden.*

Similarly to [13, 15], we use the following observation that the authors of [13] attribute to Jakub Kozik.

LEMMA 3 (see [13]). *In any proper edge coloring of G, at most $2(\Delta - 1)$ colors are 4-forbidden for any $e \in E$.*

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

Armed with Lemma 3, the general idea is to use a palette P of size $2(\Delta - 1) + Q$ colors so that whenever we (re)color an edge e, there will be at least Q colors 4-available for e (of course, coloring e may create one or more bichromatic cycles of length at least 6). At a high level, similarly to [15], our algorithm will proceed as follows:

- Start at a proper edge coloring with no bichromatic 4-cycles.

- While bichromatic cycles of length at least 6 exist, recolor the edges of one with 4-available colors.

Note that to find bichromatic cycles in a properly edge colored graph, we can just consider each of the $\binom{|P|}{2}$ pairs of distinct colors from P and seek cycles in the subgraph of the correspondingly colored edges.

6.3. Applying our framework. Given $G = (V, E)$ and a palette P of $2(\Delta - 1) + Q$ colors, let Ω be the set of all proper edge colorings of G with no monochromatic 4-cycle. Fix an arbitrary ordering π of E and an arbitrary ordering χ of P. For every even cycle C of length at least 6 in G, fix (arbitrarily) two adjacent edges e_1^C, e_2^C of C.

– Our distribution of initial state θ assigns all its probability mass to the following $\sigma_1 \in \Omega$: color the edges of E in π-order, assigning to each edge $e \in E$ the χ-greatest 4-available color. Our algorithm is as follows.

– For every even cycle C of length at least 6, we define the flaw $f_C = \{\sigma \in \Omega : C \text{ is bichromatic}\}$. Thus, a flawless $\sigma \in \Omega$ is an AEC of G.

– The set of actions for addressing f_C in state σ, i.e., $A(C, \sigma)$, comprises all $\tau \in \Omega$ that may result from the following procedure: uncolor all edges of C except for e_1^C, e_2^C; go around C, starting with the uncolored edge that is adjacent to e_2^C, etc., assigning to each uncolored edge $e \in C$ one of the 4-available colors for e at the time e is considered. Thus, by Lemma 3, $|A(C, \sigma)| \geq Q^{|C|-2}$.

LEMMA 4. *For every flaw f_C and state $\tau \in \Omega$, there is at most one arc $\sigma \xrightarrow{C} \tau$.*

Proof. Given τ and C, to recover the previous state σ it suffices to extend the bicoloring in τ of e_1^C, e_2^C to the rest of C (since C was bichromatic in σ and only edges in $C \setminus \{e_1^C, e_2^C\}$ were recolored). ∎

Thus, taking μ to be uniform and ρ such that (D, ρ, μ) is harmonic yields $\gamma_C \leq Q^{-|C|+2}$.

Let R be the symmetric directed graph with one vertex per flaw, where $f_C \rightleftarrows f_{C'}$ iff $C \cap C' \neq \emptyset$. Since a necessary condition for f_C to potentially cause $f_{C'}$ is that $C \cap C' \neq \emptyset$, we see that R is a supergraph of the causality digraph. Thus, if we run Recursive Walk with input R, to apply Theorem 2 we need to evaluate for each flaw f_C a sum over the subsets of $\Gamma_R(C)$ that are independent in R.

For $n \geq 2$, let $g(n) = \max_{e \in E} |\{(2n+2)\text{-cycles in } G \text{ that contain } e\}|$. Let $\alpha_2, \alpha_3, \ldots$ be positive numbers such that $\beta = \sum_{n=2}^{\infty} \alpha_n < \infty$. We will use $\psi_C = \psi(n) = \alpha_n / g(n)$ for a cycle C of length $2n + 2$. For a set of edges $X = \{e_1, \ldots, e_k\}$, let $\mathtt{Ind}_X$ denote the set of all k-sets of cycles $S = \{C_1, \ldots, C_k\}$, where $e_i \in C_i$ for every i, and where the cycles are edge-disjoint and, therefore, independent in R. Then,

$$\sum_{S \in \mathtt{Ind}_X} \prod_{C \in S} \psi_C \leq \left(\sum_{n=2}^{\infty} g(n)\psi(n) \right)^{|X|} = \beta^{|X|} \ .$$

Therefore, for each $(2n + 2)$-cycle C, we can bound (7) as

$$\frac{\gamma_C}{\psi_C} \sum_{S \in \mathtt{Ind}(\Gamma_R(C))} \prod_{C' \in S} \psi_{C'} \leq \frac{\gamma_C}{\psi_C} \sum_{X \subseteq C} \sum_{S \in \mathtt{Ind}_X} \prod_{C' \in S} \psi_{C'} \leq \frac{\gamma_C}{\psi_C} \sum_{X \subseteq C} \beta^{|X|} = \frac{\gamma_C}{\psi_C} (1+\beta)^{|C|} \ . \tag{20}$$

To bound (20), we observe that for arbitrary graphs, trivially, $g(n) \leq (\Delta - 1)^{2n}$. Setting $a_n = \lambda^{2n}$ and $Q = \kappa(\Delta - 1)$, where $\kappa > 0$ and $0 < \lambda < 1$ will be specified

shortly, we get

(21)
$$\frac{\gamma_C}{\psi_C}(1+\beta)^{|C|} \le \left(\frac{\Delta-1}{\lambda Q}\right)^{|C|-2}\left(1+\frac{\lambda^4}{1-\lambda^2}\right)^{|C|} = \left(\frac{1}{\lambda\kappa}\right)^{|C|-2}\left(1+\frac{\lambda^4}{1-\lambda^2}\right)^{|C|}.$$

Choosing $(\kappa,\lambda)=(2.182,0.569)$, the right-hand side of (21) becomes strictly less than 1 for every $|C|\ge 6$. Thus, for general graphs, $4.182(\Delta-1)$ colors suffice to find an AEC efficiently.

6.4. Graphs of bounded degeneracy. We will prove the following structural lemma relating degeneracy to g.

LEMMA 5. *If $G\in\mathcal{G}_d$ has maximum degree Δ, then $g(n)\le 2\binom{2n}{n}(d\Delta)^n$.*

We let $\alpha_n=\alpha\binom{2n}{n}\lambda^n$ for some $\alpha>0$ and $\lambda\in(0,\frac14)$, to be specified. Thus, $\beta=\alpha(\frac{1}{\sqrt{1-4\lambda}}-1-2\lambda)$, since $\sum_{n=0}^{\infty}\alpha_n=\alpha\sum_{n=0}^{\infty}\binom{2n}{n}\lambda^n=\frac{\alpha}{\sqrt{1-4\lambda}}$ (see [7]). Since $\frac{\gamma_C}{\psi_C}\le\frac{g(n)}{\alpha_n Q^{2n}}\le\frac{2(d\Delta)^n}{\alpha\lambda^n Q^{2n}}$ for any $(2n+2)$-cycle C, we see that (20) will be less than 1 if for every $n\ge 2$,

$$Q>\left(\frac{2(1+\beta)^2}{\alpha}\right)^{1/2n}\cdot\frac{1+\beta}{\sqrt{\lambda}}\cdot\sqrt{d\Delta}\ .$$

If we take α,λ such that $2(1+\beta)^2<(1-\delta)\alpha$, then taking $Q>\frac{1+\beta}{\sqrt{\lambda}}\cdot\sqrt{d\Delta}$ satisfies (7). In particular, taking $(\alpha,\lambda)=(2.76,0.086)$ works, in which case $\frac{1+\beta}{\sqrt{\lambda}}<4$. Regarding the running time, it can easily be seen that that T_0 is a polynomial in $|E|$, Δ, and the number of colors used (since δ is a constant).

Proof of Lemma 5. Fix any edge $e=\{u,v\}\in E$. To enumerate the $(2n+2)$-cycles containing e, we will partition them into equivalence classes as follows. First, we orient all edges of G arbitrarily to get a digraph D. Consider now the two possible traversals of the path $C\setminus\{u,v\}$, i.e., one starting at u and the other starting at v. For each traversal, generate a string in $\{0,1\}^{2n}$ whose characters correspond to successive vertices of the path, other than the endpoints, and denote whether the corresponding vertex was entered along an edge oriented in agreement (1) or in disagreement (0) with the direction of travel. Observe that each $2n-1$ edge of C that has no vertex from $\{u,v\}$ will create a 1 in one string and a 0 in the other. Therefore, at least one of the two strings will have at least $\lceil(2n-1)/2\rceil=n$ 1's. Select that string, breaking ties in favor of the string corresponding to starting at u. Then, in the selected string, convert as many of the leftmost 1's as needed to 0's, so that the resulting string has exactly n 1's. Finally, prepend a single bit to indicate whether the winning traversal started at u or to v. The resulting string is the representative of C's equivalence class. Clearly, there are at most $2\binom{2n}{n}$ equivalence classes.

To bound the number of cycles in a class, we enumerate the possibilities for the $2n$ vertices other than u,v as follows. After reading the bit indicating whether the cycles in the class start at u or at v, we interpret each successive character of the representative string to indicate whether we can choose from among the out-neighbors or from among all neighbors of the current vertex. By the string's construction, there will be exactly n choices of each kind, and therefore the total number of choices will be at most $\mathrm{Out}^n\Delta^n$, where Out is an upper bound on the out-degree of D.

To conclude the argument, we note that since $G\in\mathcal{G}_d$, we can direct its edges so that every vertex has out-degree at most d by repeatedly removing any vertex v of

current degree at most d (it always exists) and, at the time of removal, orienting its current neighbors away from v. ∎

Acknowledgments. We are grateful to Dan Kral for providing us with Lemma 5, and to Louis Esperet for pointing out an error in our application of Theorem 2 to yield Theorem 6 in a previous version of the paper. F.I. is grateful to Alistair Sinclair for many fruitful conversations.

REFERENCES

[1] D. Achlioptas and A. Coja-Oghlan, *Algorithmic barriers from phase transitions*, in Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Press, Piscataway, NJ, 2008, pp. 793–802.

[2] D. Achlioptas and F. Iliopoulos, *Focused stochastic local search and the Lovász local lemma*, in Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), R. Krauthgamer, ed., SIAM, Philadelphia, 2016, pp. 2024–2038, https://doi.org/10.1137/1.9781611974331.ch141.

[3] D. Achlioptas and F. Iliopoulos, *Random walks that find perfect objects and the Lovász local lemma*, J. ACM, 63 (2016), 22.

[4] M. Alava, J. Ardelius, E. Aurell, P. Kaski, S. Krishnamurthy, P. Orponen, and S. Seitz, *Circumspect descent prevails in solving random constraint satisfaction problems*, Proc. Natl. Acad. Sci. USA, 105 (2008), pp. 15253–15257.

[5] N. Alon, *A parallel algorithmic version of the local lemma*, Random Structures Algorithms, 2 (1991), pp. 367–378.

[6] R. Bissacot, R. Fernández, A. Procacci, and B. Scoppola, *An improvement of the Lovász local lemma via cluster expansion*, Combin. Probab. Comput., 20 (2011), pp. 709–719.

[7] K. Boyadzhiev, *Series with central binomial coefficients, Catalan numbers, and harmonic numbers*, J. Integer Sequences, 15 (2012), 12.1.7.

[8] J. D. Catarata, S. Corbett, H. Stern, M. Szegedy, T. Vyskocil, and Z. Zhang, *The Moser-Tardos resample algorithm: Where is the limit? (An experimental inquiry)*, in Proceedings of the Nineteenth Workshop on Algorithm Engineering and Experiments (ALENEX) S. Fekete and V. Ramachandran, eds., SIAM, Philadelphia, 2017, pp. 159–171, https://doi.org/10.1137/1.9781611974768.13.

[9] T. F. Coleman and J.-Y. Cai, *The cyclic coloring problem and estimation of sparse Hessian matrices*, SIAM J. Algebraic Discrete Methods, 7 (1986), pp. 221–235, https://doi.org/10.1137/0607026.

[10] T. F. Coleman and J. J. Moré, *Estimation of sparse Hessian matrices and graph coloring problems*, Math. Programming, 28 (1984), pp. 243–270.

[11] V. Dujmović, G. Joret, J. Kozik, and D. R. Wood, *Nonrepetitive colouring via entropy compression*, Combinatorica, 36 (2016), pp. 661–686.

[12] P. Erdős and J. Spencer, *Lopsided Lovász local lemma and Latin transversals*, Discrete Appl. Math., 30 (1991), pp. 151–154.

[13] L. Esperet and A. Parreau, *Acyclic edge-coloring using entropy compression*, European J. Combin., 34 (2013), pp. 1019–1027.

[14] G. Fertin, A. Raspaud, and B. Reed, *Star coloring of graphs*, J. Graph Theory, 47 (2004), pp. 163–182.

[15] I. Giotis, L. M. Kirousis, K. I. Psaromiligkos, and D. M. Thilikos, *Acyclic edge coloring through the Lovász local lemma*, Theor. Comput. Sci., 665 (2017), pp. 40–50.

[16] J. Grytczuk, J. Kozik, and P. Micek, *New approach to nonrepetitive sequences*, Random Structures Algorithms, 42 (2013), pp. 214–225.

[17] B. Haeupler, B. Saha, and A. Srinivasan, *New constructive aspects of the Lovász local lemma*, J. ACM, 58 (2011), 28.

[18] D. G. Harris and A. Srinivasan, *A constructive algorithm for the Lovász local lemma on permutations*, in Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA) C. Chekuri, ed., SIAM, Philadelphia, 2014, pp. 907–925, https://doi.org/10.1137/1.9781611973402.68.

[19] N. J. A. Harvey and J. Vondrák, *An algorithmic proof of the Lovász local lemma via resampling oracles*, in Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS) V. Guruswami, ed., IEEE Press, Piscataway, NJ, 2015, pp. 1327–1346.

[20] F. Iliopoulos, *Commutative algorithms approximate the LLL-distribution*, in Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018), E. Blais, K. Jansen, J. D. P. Rolim, and D. Steurer, eds., LIPIcs 116, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2018, 44.

[21] T. R. Jensen and B. Toft, *Graph Coloring Problems*, Wiley Ser. Discrete Math. Optim., Wiley, New York, 2011.

[22] K. B. R. Kolipaka and M. Szegedy, *Moser and Tardos meet Lovász*, in Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC '11), ACM, New York, 2011, pp. 235–244.

[23] V. Kolmogorov, *Commutativity in the algorithmic Lovász local lemma*, in Proceedings of the 57th Annual IEEE Annual Symposium on Foundations of Computer Science (FOCS), I. Dinur, ed., IEEE Press, Piscataway, NJ, 2016, pp. 780–787.

[24] A. V. Kostochka, E. Sopena, and X. Zhu, *Acyclic and oriented chromatic numbers of graphs*, J. Graph Theory, 24 (1997), pp. 331–340.

[25] M. Molloy, *The list chromatic number of graphs with small clique number*, J. Combin. Theory Ser. B, 134 (2019), pp. 264–284.

[26] M. Molloy and B. Reed, *Further algorithmic aspects of the local lemma*, in Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing (STOC '98), ACM, New York, 1998, pp. 524–529.

[27] R. A. Moser, *A constructive proof of the Lovász local lemma*, in Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC '09), ACM, New York, 2009, pp. 343–350.

[28] R. A. Moser and G. Tardos, *A constructive proof of the general Lovász local lemma*, J. ACM, 57 (2010), 11.

[29] S. Ndreca, A. Procacci, and B. Scoppola, *Improved bounds on coloring of graphs*, European J. Combin., 33 (2012), pp. 592–609.

[30] C. H. Papadimitriou, *On selecting a satisfying truth assignment*, in Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Press, Piscataway, NJ, 1991, pp. 163–169.

[31] W. Pegden, *Highly nonrepetitive sequences: Winning strategies from the local lemma*, Random Structures Algorithms, 38 (2011), pp. 140–161.

[32] W. Pegden, *An extension of the Moser–Tardos algorithmic local lemma*, SIAM J. Discrete Math., 28 (2014), pp. 911–917, https://doi.org/10.1137/110828290.

[33] B. Selman, H. A. Kautz, and B. Cohen, *Local search strategies for satisfiability testing*, in Proceedings of the Second DIMACS Challenge on Cliques, Coloring, and Satisfiability, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 26, D. S. Johnson and M. A. Trick, eds., AMS, Providence, RI, 1996, pp. 521–532.

[34] J. B. Shearer, *On a problem of Spencer*, Combinatorica, 5 (1985), pp. 241–245.

[35] M. Szegedy, *The Lovász local lemma—a survey*, in Computer Science—Theory and Applications: Proceedings of the 8th International Computer Science Symposium in Russia (CSR 2013), A. A. Bulatov and A. M. Shur, eds., Lecture Notes in Comput. Sci. 7913, Springer, Heidelberg, 2013, pp. 1–11.