Stochastic Control via Entropy Compression*

Dimitris Achlioptas[†1], **Fotis Iliopoulos**[‡2], **and Nikos Vlassis**[3]

1 **Department of Computer Science, UC Santa Cruz, Santa Cruz, CA, USA**
optas@cs.ucsc.edu
2 **Department of Electrical Engineering and Computer Science, UC Berkeley, Berkeley, CA, USA**
fotis.iliopoulos@berkeley.edu
3 **Adobe Research, San Jose, CA, USA**
nikos.vlassis@gmail.com

Abstract

Consider an agent trying to bring a system to an acceptable state by repeated probabilistic action. Several recent works on algorithmizations of the Lovász Local Lemma (LLL) can be seen as establishing sufficient conditions for the agent to succeed. Here we study whether such stochastic control is also possible in a noisy environment, where both the process of state-observation and the process of state-evolution are subject to adversarial perturbation (noise). The introduction of noise causes the tools developed for LLL algorithmization to break down since the key LLL ingredient, the sparsity of the causality (dependence) relationship, no longer holds. To overcome this challenge we develop a new analysis where entropy plays a central role, both to measure the rate at which progress towards an acceptable state is made and the rate at which noise undoes this progress. The end result is a sufficient condition that allows a smooth tradeoff between the intensity of the noise and the amenability of the system, recovering an asymmetric LLL condition in the noiseless case.

1998 ACM Subject Classification G.3 Probabilistic Algorithms, I.2.8 Control Theory

Keywords and phrases Stochastic Control, Lovász Local Lemma

Digital Object Identifier 10.4230/LIPIcs.ICALP.2017.83

1 Introduction

Consider a system with a large state space Ω, hidden from view inside a box. On the outside of the box there are lightbulbs and buttons. Each lightbulb corresponds to a set $f_i \subseteq \Omega$ and is lit whenever the current state of the system is in f_i. We think of each set f_i as containing all states sharing some negative feature $i \in [m]$ and refer to each such set as a *flaw*, letting $F = \{f_1, f_2, \ldots, f_m\}$. For example, if the system corresponds to a graph G with n vertices each of which can take one of q colors, then $\Omega = [q]^n$, and we can define for each edge e_i of G the flaw f_i to contain all assignments of colors to the vertices of G that assign the same color to the endpoints of e_i. Following linguistic convention, instead of mathematical, we will say that flaw f is present in state σ whenever $f \ni \sigma$ and that state σ is *flawless* if no flaw is present in σ. The buttons correspond to actions, i.e., to mechanisms for state evolution. Specifically, taking action a while in state σ moves the system to a new state τ, selected from a probability distribution that depends on both σ and a.

* A full version of this paper appears as [1], https://arxiv.org/abs/1607.06494.
† Research supported by NSF grant CCF-1514128.
‡ Research supported by NSF grant CCF-1514434. Part of of this work was done while at Adobe Research.

44th International Colloquium on Automata, Languages, and Programming (ICALP 2017).
Editors: Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl;
Article No. 83; pp. 83:1–83:13


Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Outside the box, an agent called the *controller* observes the lightbulbs and pushes buttons, in an effort to bring the system to a flawless state. Specifically, if $O(\sigma) \in \{0,1\}^m$ denotes the lightbulb bitvector, with 1 corresponding to lit, the controller repeatedly applies a function P, called a *policy*, that maps $O(\sigma)$ to a distribution over actions. Thus, overall, state evolution proceeds as follows: if the current (hidden) state is $\sigma \in \Omega$, the controller observes $O(\sigma)$ and samples an action from $P(O(\sigma))$; after she takes the chosen action, the system, internally and probabilistically, moves to a new (hidden) state τ, selected from a distribution that depends on both σ and the action taken.

Our work begins with the observation that several recent results [23, 24, 18, 14, 3, 15, 2, 19] on LLL algorithmization can be seen as giving sufficient conditions for a controller as above to be able to bring the system to a flawless state quickly, with high probability. Motivated by this viewpoint we ask if conditions for LLL algorithmizations can be seen as *stability criteria* and give results for more general settings, e.g., Partially Observable Markov Decision Processes (POMDPs). Given the capacity of LLL algorithmization arguments to establish convergence in highly non-convex domains, a major pain point in control theory, we believe that bringing such arguments to stochastic control is a first step in a fruitful direction. In order to move in that direction we generalize the setting described so far in two ways:

- The mapping O from states to observations is *stochastic*: the lightbulbs are unreliable, exhibiting both false-positives and false-negatives.
- Both the environment surrounding the system and the implementation of actions are *noisy*: the controller is not the only agent affecting state evolution and flaws may be introduced into the state for reasons unrelated to her actions, even spontaneously.

Naturally, the question is whether sufficient conditions for quick convergence to flawless states can still be established in this setting. We answer the question affirmatively and show, in a precise mathematical sense, that the less internal conflict there is in the system, the more noise the controller can tolerate. In order to prove this we require the controller to be *focused* and to *prioritize*. That is, we will assume that the flaws are ordered by priority according to an arbitrary but fixed permutation π of F, and we will ascribe the action taken by the controller in each step to the present flaw (focus) of highest priority (prioritization). The analysis will then take into account both how good the actions are at ridding the state of that flaw and how damaging they are in terms of introducing new flaws. In particular, with this attribution mechanism in place, and similarly to LLL algorithmization arguments, we will say that flaw f_i can cause flaw f_j if there exists a state transition with non-zero probability under the policy, from a state in which f_i is the highest priority flaw and f_j is absent, to a state in which f_j is present.

The main challenge we face is that in the presence of noise the causality relationship becomes dense. To overcome this we develop a new analysis in which causality is not a binary relationship, but one weighted by the *frequency* of interactions. In particular, our condition guaranteeing that the controller will succeed within a reasonable amount of time allows the causality graph to become arbitrarily dense, if the frequency of interactions is sufficiently small. Turning the sparsity of the causality relationship into a *soft* requirement is a major departure from the LLL setting and our main technical contribution. We do this by developing an entropy compression argument, in which we carefully amortize the entropy injected into the system to encode the effect of noise on the state trajectory. It is worth pointing out that even though our technique applies to the far more general noisy setting, in the absence of noise it recovers the main result of [3], thus providing a smooth relationship between lack of internal conflict and robustness to noise.

2 Formal Setting and Statement of Results

In the absence of observational and environmental noise we can think of the state evolution under a policy P as a random walk on a certain digraph on Ω. Specifically, at each flawed state $\sigma \in \Omega$, for each action in the support of $P(O(\sigma))$, there is a bundle of outgoing arcs of total probability 1, corresponding to the state-transitions from σ under this action. The convolution of $P(O(\sigma))$ with the distribution inside each bundle yields the state-transition probability distribution from each flawed state σ.

The presence of observational and environmental noise both distorts the transition probabilities and introduces new transitions. For example, whenever observational noise causes $O(\sigma)$ to differ from the set of flaws truly present in σ, the controller may chose an action (from the support of $P(O(\sigma))$) under which there are transitions from σ that were not present in the noise-free digraph. We model the overall distortion induced by noise by taking the noise-free digraph, which we think of as the *principal* mechanism for state evolution, reducing the probabilities on all its edges uniformly by a factor of $1-p$, and allowing the leftover probability mass to be distributed arbitrarily, in order to form the noise. More precisely:

- Let D_{po} be the digraph on Ω of possible state-transitions under policy P, with a self-loop added at every flawless state. Let ρ_{po} be the P-induced state-transition probability distribution, augmented so that all self-loops at flawless states have probability 1.
- Let D_{ns} be an **arbitrary** digraph on Ω. For each vertex σ in D_{ns}, let $\rho_{\text{ns}}(\sigma,\cdot)$ be an **arbitrary** probability distribution on the arcs leaving σ.
- We will analyze the Markov chain on Ω which at every $\sigma \in \Omega$, with probability p follows an arc in D_{ns} and with probability $1-p$ follows an arc in D_{po}. Formally, for every $\sigma \in \Omega$,

$$\rho(\sigma,\cdot) = (1-p)\cdot\rho_{\text{po}}(\sigma,\cdot) + p\cdot\rho_{\text{ns}}(\sigma,\cdot) \ .$$

  We assume that the system starts at a state σ_1, according to some unknown probability distribution θ.

Requiring that the effect of noise is captured by a mixture of the original (principal) chain and an arbitrary chain is the only assumption that we make. In particular, by allowing D_{ns} and ρ_{ns} to be arbitrary we forego the need to posit specific models of observational and environmental noise, lending greater generality to our results. To see this, let $U(\sigma)$ denote the set of flaws actually present in σ (and, slightly abusing notation, also the characteristic vector of $U(\sigma) \subseteq F$). In any step where the state transition distribution is not the principal one, we can think of this as occurring because $O(\sigma) \neq U(\sigma)$ and the distribution corresponds to $P(O(\sigma))$, or because $O(\sigma) \neq U(\sigma)$ and the distribution does not *even* correspond to $P(O(\sigma))$, or because $O(\sigma) = U(\sigma)$ but, silently, the distribution followed is different from $P(O(\sigma))$. In particular, notice that whenever $O(\sigma) = \mathbf{0}$, the controller thinks she has arrived at a flawless state and, thus, authorizes a self-loop with probability 1. In such a case, the fact that the system will follow ρ_{ns} with probability p means that we are allowing the noise not only to trick the controller to inactivity but also to silently move the system to a new state. Similarly, after the system arrives at a flawless state, i.e., $U(\sigma) = \mathbf{0}$, with probability p it will then follow an arc in D_{ns}, potentially to a flawed state. We allow this to occur to be consistent with (i) the idea that observational noise can occur at any state, even a flawless one, thus causing unneeded, potentially detrimental action, and (ii) with the idea that flaws can be introduced spontaneously from the environment at any state. Our goal is, thus, to prove that from *any* initial state, after a small number of steps, the system will

reach a flawless state, despite the noise. As we will see, what will matter about the noise is the extent to which noise-induced transitions introduce flaws in the state.

Let $D = D_{\rm po} \cup D_{\rm ns}$. To avoid certain trivialities we will assume that there exists a constant $B < \infty$ such that $2^{-B} < \rho(\sigma, \tau) < 1 - 2^{-B}$ for every arc $(\sigma, \tau) \in D$. For each state σ, we denote the highest priority flaw present in σ by $\pi(\sigma)$; if $\pi(\sigma) = f_i$, we label all arcs leaving σ as $\sigma \xrightarrow{i} \cdot$, i.e., with the index of the flaw to which we attribute the transition (we use i instead of f_i as the label to lighten notation). We will refer to $\pi(\sigma)$ as the flaw *addressed* at σ.

▶ **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 $f_j \not\ni \sigma$. The digraph on $[m]$ where $i \to j$ iff D contains an arc such that f_i causes f_j is the causality digraph $C(D)$.*

▶ **Neighborhood.** *The* neighborhood *of a flaw f_i in $C = C(D)$ is $\Gamma(f_i) = \{f_i\} \cup \{f_j : i \to j$ exists in $C\}$.*

For our condition we will need to bound from *below* the entropy injected into the system in each step. To that end we define the potential of each flaw f_i to be

$$\text{Potential}(f_i) = \min_{\sigma:\pi(\sigma)=f_i} H[\rho(\sigma, \cdot)] \ . \tag{1}$$

We extend the definition to sets of flaws i.e., $\text{Potential}(S) = \sum_{f \in S} \text{Potential}(f)$, where $\text{Potential}(\emptyset) = 0$.

In the absence of noise, $\text{Potential}(f_i)$ expresses a lower bound on the diversity of ways to address flaw f_i, by bounding from below the "average number of random bits consumed" whenever f_i is addressed. Thus, it bounds from below the rate at which the controller explores the state space *locally*. The presence of noise may decrease or may increase the potential. For example, if all arcs in $D_{\rm ns}$ are self-loops, then the noise is equivalent to the action-buttons "sometimes not working" and its only (and very benign) effect is to slow down the exploration by a constant factor. At the other extreme, if $D_{\rm ns}$ is the complete digraph on Ω and $\rho_{\rm ns}$ is uniform, then (unless p is extremely small) the situation is, clearly, hopeless. Correspondingly, even though the potential has been greatly increased, the causality relationship is complete. We note that, trivially, the potential of each flaw is bounded from below by the minimum entropy injected by the principal alone whenever the flaw is addressed, i.e., $\text{Potential}(f_i) \geq (1-p) \min_{\sigma:\pi(\sigma)=f_i} H[\rho_{\rm po}(\sigma)]$.

The other important characteristic of each flaw f_i is its congestion, i.e., the maximum number of arcs with label i that lead to the same state. For the same reason we would like the potential of a flaw to be big, we would like its congestion to be small: if arcs from different states in f_i lead to the same state, then exploration slows down and the entropy injected into the system must be appropriately discounted in order to yield a good measure of the rate of state space exploration. To see this observe that $\text{Potential}(f_i)$ is independent of the destinations of the arcs leaving f_i and compare the case where these destinations are all distinct with the case where they all lie in a small (bottleneck) set. As the congestion due to the principal and the congestion due to noise will have different effects, we need to account for them separately. Let $A_{\rm po}(\sigma)$ denote the support of $\rho_{\rm po}(\sigma, \cdot)$ and $A_{\rm ns}(\sigma)$ denote the support of $\rho_{\rm ns}(\sigma, \cdot)$.

▶ **Congestion.** *For any flaw $f_i \in F$, let*

$$\begin{aligned} \text{Congestion}_{\rm po}(f_i) &= \max_{\tau \in \Omega} |\{\sigma \in f_i : \tau \in A_{\rm po}(\sigma)\}| \\ \text{Congestion}_{\rm ns}(f_i) &= \max_{\tau \in \Omega} |\{\sigma \in f_i : \tau \in A_{\rm ns}(\sigma)\}| \ . \end{aligned}$$

Let $\mathrm{b}_{\mathrm{po}}^{f_i} = \log_2 \mathrm{Congestion}_{\mathrm{po}}(f_i)$. *Let* $\mathrm{b}_{\mathrm{ns}}^{f_i} = \log_2 \mathrm{Congestion}_{\mathrm{ns}}(f_i)$. *Let* $\mathrm{b}_{\mathrm{ns}} = \max_{f_i \in F} \mathrm{b}_{\mathrm{ns}}^{f_i}$.

Let C_{po} and C_{ns} be the causality graphs of D_{po} and D_{ns}, respectively, and let $\Gamma_{\mathrm{po}}(f_i)$ and $\Gamma_{\mathrm{ns}}(f_i)$ be the corresponding neighborhoords. Let $\Delta_i = |\Gamma_{\mathrm{ns}}(f_i)|$. Recall that $h(p) = -p\log_2 p - (1-p)\log_2(1-p)$ is the *binary entropy* of $p \in [0,1]$. To express the lost efficiency due to noise in addressing flaw f_i, we let

$$\begin{aligned} q_i(p) &= p\left(\Delta_i\left(\mathrm{b}_{\mathrm{ns}} + \frac{5}{2} + h(p)\right) - 2 - h(p)\right) \\ &\leq p\,\Delta_i(\mathrm{b}_{\mathrm{ns}} + 4) \ . \end{aligned}$$

Observe that $q_i(p)$ is independent of the policy and that its leading term is $p\Delta_i$. This means that, unlike the LLL, the number, Δ_i, of different flaws that may be introduced when addressing a flaw can be arbitrarily large if the *frequency of interactions* between flaws, captured by p, is sufficiently small. Our main result establishes a condition under which the probability of not reaching a flawless state within $O(\log_2|\Omega| + m)$ steps is exponentially small. To state it define for each flaw f_i,

$$\mathrm{Amenability}(f_i) = \mathrm{Potential}(f_i) - \mathrm{b}_{\mathrm{po}}^{f_i} \ .$$

▶ **Theorem 1.** *If for every flaw* $f_i \in F$,

$$\sum_{f_j \in \Gamma_{\mathrm{po}}(f_i)} 2^{-\mathrm{Amenability}(f_j) + q_j(p)} < 2^{-(2+h(p))} \ , \tag{2}$$

then there exists a constant $R > 0$ *depending on the slack in* (2)*, such that for every* $s > 1/2$*, the probability of not reaching a flawless state after* $Rs(\log_2|\Omega| + m)$ *steps is less than* $\exp(-s)$.

▶ Remark. In the noiseless case, i.e., when $p = 0$, equation (2) becomes an asymmetric LLL criterion. In particular, the main result of [3] is that if $\mathrm{b}_{\mathrm{po}}^{f_i} = 0$ and all distributions $\rho_{\mathrm{po}}(\sigma, \cdot)$ are uniform over their support $A_{\mathrm{po}}(\sigma)$, then, a sufficient condition for reaching a flawless state quickly is that for every $f_i \in F$, $\sum_{f_j \in \Gamma_{\mathrm{po}}(f_i)} 1/a_j < 1/\mathrm{e}$, where $a_j = \min_{\sigma \in f_j : \pi(\sigma) = f_j} |A_{\mathrm{po}}(\sigma)|$. We see that in this setting our condition (2) recovers this, up to the constant on the right hand side, i.e., $1/4$ vs. $1/\mathrm{e}$.

3 Related Work

3.1 POMDPs and the Reachability Problem

Markov Decision Processes (MDPs) are widely used models for describing problems in stochastic dynamical systems [13, 28, 7], where an agent repeatedly takes actions to achieve a specific goal while the environment reacts to these actions in a stochastic way. In an MDP the agent is assumed to be able to *perfecty* observe the current state of the system and take action based on her observations. In a *partially* observable Markov Decision Process (POMDP) the agent only receives limited, and possibly inaccurate, information about the current state of the system. POMDPs have been used to model and analyze problems in artificial intelligence and machine learning such as reinforcement learning [9, 17], planning under uncertainty [16], etc.

Formally, a discrete POMDP is defined by the following primitives (all sets are assumed finite): (i) a *state space* Ω, (ii) a finite alphabet of *actions* $\mathcal{A}$, (iii) an observation space $\mathcal{O}$, (iv) an *action*-conditioned state transition model $\Pr(\tau|\sigma, a)$, where $\sigma, \tau \in \Omega$ and $a \in \mathcal{A}$,

(v) an observation model $\Pr(o|\sigma)$, where $\sigma \in \Omega$ and $o \in \mathcal{O}$, (vi) a cost function $\mathrm{c} : \Omega \mapsto \mathbb{R}$ (or more generally a map from state-action pairs to the reals), and (vii) a desired criterion to minimize, e.g., expected total cumulative cost $\sum_{t=0}^{\infty} \mathbb{E}\left[\mathrm{c}(\sigma_t)\right]$, where σ_t is the random variable that equals the t-th state of the trajectory of the agent. Finally, various choices of controllers are possible. For instance, a *stochastic memoryless* controller is a map from the *current* observation to a probability distribution over actions, whereas a *belief-based* controller conditions its actions on probability distributions over the state space (i.e., beliefs) that are sequentially updated (using Bayes rule or some approximation of it) while the agent is interacting with the environment.

Unfortunately, the problem of computing an optimal policy for a POMDP, i.e., designing a controller that minimizes the expected cost, is highly intractable [27, 25] and, in general, undecidable [21]. Notably, the problem remains hard even if we severely restrict the class of controllers over which we optimize [27, 20, 12, 31]. As far as we know, the only tractable case [31] requires both the cost function and the class of controllers over which we optimize to be extremely restricted. In particular, the controller can not observe or remember anything and must apply the same distribution over actions in every step.

An important special case that has motivated our work is the *reachability* problem for POMPDs. Here, one has a set of *target* states $T \subseteq \Omega$, and the goal is to design a controller that starting from a distribution θ over Ω, guides the agent to a state in T (almost surely) with the optimal expected total cumulative cost. As shown in [8], the problem is undecidable in the general case. In the same work, for the case where the costs are positive integers and the observation model is deterministic, i.e., the observations induce a partition of the state space, the authors give an algorithm which runs in time doubly-exponential in $|\Omega|$ and returns doubly-exponential lower and upper bounds for the optimal expected total cumulative cost, using a belief-based controller. On the other hand, our work establishes a sufficient condition for a stochastic memoryless controller to reach the target set T rapidly (in time logarithmic in $|\Omega|$ and linear in $|F|$), in the case where each individual observation is binary valued (set membership) and the observation model is arbitrarily stochastic. To our knowledge, this is the first tractability result for a nontrivial class of POMDPs under stochastic memoryless controllers.

3.2 Focusing and Prioritization

To achieve our results the controller must be focused and prioritize. The idea of focusing was introduced by Papadimitriou [26] in the context of satisfiability algorithms, and amounts to "if it ain't broken don't fix it", i.e., state evolution should only happen by changing the values of variables that participate in at least one violated constraint. One way to implement this idea is to always first select a violated constraint (flaw) and then take actions that tend to get rid of it. This has been an extremely successful idea in practice [29, 4] and it is often materialized by selecting a *random* flaw to address in each step. We remark that our methods allow, in fact, also the analysis of controllers that address a random flaw in each step, but for simplicity of exposition we chose to only present the case of a fixed permutation (prioritization).

Focusing is not only a good algorithmic idea, but also enables proofs of termination. Specifically, at the foundation of the argument of Moser and Tardos [24] is the following observation: whenever an algorithm (focused or not) takes t or more steps to reach a flawless state, say through flawed states $\sigma_1, \sigma_2, \ldots, \sigma_t$, there exists, by definition, a sequence of flaws $w_1, w_2, \ldots, w_t$ such that $\sigma_i \in w_i$. Therefore, by establishing a (potentially randomized) rule for selecting a flaw present in the state at each step, we can construct a random variable

$W_t = w_1, w_2, \ldots, w_t$ to act as a *witness* of the fact that the algorithm took at least t steps. While, though, prima facie all constructions are equivalent, our capacity to bound the set of all possible such sequences is not. In particular, if the algorithm is focused and in each step we report the flaw on which the algorithm focused, then we can take advantage of the following observation: each appearance of a flaw f_i in the witness sequence, with the potential exception of the very first, must be preceded by a distinct appearance of a flaw f_j that causes f_i. This allows us to bound the rate at which the entropy of the set of t-witness sequences grows with t. Of course, in a general setting, there is good reason to believe that prioritization, i.e., focusing on the flaw determined by a fixed permutation, will be not be the best one can do. In particular, observe that for the same $D_{\rm po}$, different permutations π give rise to different causality graphs. On the other hand, at the level of generality of this work, i.e., without any assumptions about the system at hand, we can not really hope for a more intelligent choice.

3.3 LLL algorithmization

The Lovász Local Lemma (LLL) [11] is a non-constructive method for proving the *existence* of flawless states that has served as a cornerstone of the probabilistic method. To use the LLL one provides a probability measure μ on Ω, often the uniform measure, transforming flaws to ("bad") events, so that the existence of flawless states is equivalent to $\mu(\bigcup_{i=1}^{m} f_i) < 1$. The key quantity to control in order to prove this is negative dependence, i.e., the extent to which the probability of a bad event may be increased (boosted) by conditioning on the non-occurrence of other bad events. Roughly speaking, the LLL requires that for each bad event f, only a small number of other bad events should be able to boost $\mu(f)$ in this manner, whereas conditioning on the non-occurrence of all other bad events should not increase $\mu(f)$. Representing the boosting relationship in a graphical manner, with vertices corresponding to bad events pointing to their potential boosters, at a high level, the LLL requirement is that this digraph is sparse.

As one can imagine, whenever one proves that Ω contains flawless objects via the LLL it is natural to then ask if some such object can be found efficiently. Making the LLL constructive has been a long quest, starting with the work of Beck [6], with subsequent works of Alon [5], Molloy and Reed [22], Czumaj and Scheideler [10], Srinivasan [30] and others. Each of these works established a method for finding flawless objects efficiently, but with additional conditions relative to the LLL. A breakthrough was made by Moser [23] who gave a very elegant algorithmization of the LLL for satisfiability via entropy compression. Very shortly afterwards, Moser and Tardos in a landmark paper [24] made the LLL constructive for every product measure μ. Specifically, they proved that if one starts by sampling an initial state according to μ, and in every step selects an arbitrary occurring bad event and resamples its variables according to μ, then with high probability a flawless state will be reached within $O(m)$ steps.

Following [24], several works [18, 14, 3, 15, 2, 19] have extended the scope of LLL algorithmization beyond product measures. In these works, unlike [24], one has to also provide either an explicit algorithm [18, 14], or an algorithmic framework [3, 2, 15, 19], along with a way to capture the *compatibility* between the algorithm's actions for addressing each flaw f_i and the measure μ. As was shown in [15, 2, 19], one can capture compatibility by letting

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

where $\nu_i(\tau)$ is the probability of ending up at state τ at the end of the following experiment: sample $\sigma \in f_i$ according to μ and address flaw f_i at σ. An algorithm achieving $d_i = 1$ is a *resampling oracle* for flaw f_i. If $d_i = 1$ for every $i \in [m]$, then it was proven in [15] that the causality digraph equals the boosting digraph mentioned above and the condition for success is identical to that of the LLL (observe that the resampling algorithm of Moser and Tardos [24] is trivially a resampling oracle for every flaw). More generally, ascribing to each flaw f_i the *charge* $\gamma(f_i) = d_i \cdot \mu(f_i)$, yields the following user-friendly algorithmization condition [2], akin to the asymmetric Local Lemma: if for every flaw $f_i \in F$,

$$\sum_{f_j \in \Gamma(f_i)} \gamma(f_j) < \frac{1}{4} \ , \tag{4}$$

then with high probability the algorithm will reach a sink after $O(\log |\Omega| + m)$ steps.

Even though the noiseless case is only tangential to the main point of this work, as an indication of the sharpness of our analysis, we point out that in the noiseless case, our condition (2) is identical to (4) with $\gamma(f_i)$ replaced by $\chi(f_i) := 2^{-\text{Potential}(f_i)+b^{f_i}_{\text{po}}}$. In general, $\gamma(f_i)$ and $\chi(f_i)$ are incomparable. Roughly speaking, settings where $b^{f_i}_{\text{po}}$ is small and d_i is large favor $\chi(f_i)$ over $\gamma(f_i)$ and vice versa, while the two meet when $b^{f_i}_{\text{po}} = 0$, μ is uniform, and the transition probabilities are uniform, as in [3].

In terms of techniques, as hinted in Section 3.2, proofs of LLL algorithmizations consist of two independent parts. In one part, one bounds from above the probability of any witness sequence occurring, or in the case of Moser's entropic argument, bounds from below the entropy injected to the system while addressing the sequence. In the other part, one has to estimate the [entropy of the] set of possible witness sequences, using syntactic properties mandated by causality: roughly speaking every occurrence of a flaw in a witness sequence, with the potential exception of the very first, must be preceded by an occurrence of some flaw that causes it. Finally, one compares the rate at which the probability of a t-step witness sequence decreases (or the rate at which entropy is increased) with the rate at which the [entropy of the] set of possible witness sequences increases, to establish that their product tends to 0 with t.

In this paper, exactly because we aim to capture the intensity of interactions between flaws under adversarial noise, we need to take a different approach. In particular, our proof can be thought of as entangling the two parts described above in order to establish that, while adversarial noise can make the imposed syntactic requirements inherited by the causality graph very weak (by making the graph extremely dense), the fact that the intensity of the noise is low, suffices to control the growth rate of the entropy of the set of witness sequences. The result is a carefully tuned argument that amortizes the entropy injected into the system against its effect on the entropy of the set of Break Forests. Key to the capacity to perform this amortization is the use of so-called Break Forests, introduced in [3], which localize in time the introduction of new flaws in the state. This property of Break Forests was not used in earlier works [3, 2] and allows us to use a different amortization for the flaws introduced by the principal vs. those introduced by noise.

4 Termination via Compression

Our analysis will not depend in any way on the distribution θ of the initial state. As a result, without loss of generality, we can assume that the process starts at an arbitrary but fixed state σ_{init}. We let $A(\sigma)$ denote the support of $\rho(\sigma, \cdot)$, i.e., $A(\sigma)$ is the set of all states reachable by the process in a single step from σ.

▶ **Definition 2.** We refer to the (random) sequence $\sigma_{\text{init}} = \sigma_1, \ldots, \sigma_{t+1}$, entailing the first t steps of the process, as the *t-trajectory*. A t-trajectory is *bad* iff $\sigma_1, \ldots, \sigma_{t+1}$ are all flawed.

We model the set of all possible trajectories as an infinite tree whose root is labelled by $\sigma_1 = \sigma_{\text{init}}$. The root has $|A(\sigma_{\text{init}})|$ children corresponding to (and labelled by) each possible value of σ_2. More generally, a vertex labeled by σ has $|A(\sigma)|$ children, each child labeled by a distinct element of $A(\sigma)$, i.e., a distinct possible value of σ_{i+1}. Every edge of this infinite vertex-labelled tree is oriented away from the root and labelled by the probability of the corresponding transition, i.e., $\rho(\sigma, \tau)$, where σ is the parent and τ is the child vertex. By our assumption, every such edge label is at least 2^{-B}.

We call the above labelled infinite tree the *process tree* and note that it is nothing but the unfolding of the Markov chain corresponding to the state-evolution of the process. In particular, for every vertex v of the tree, the probability, p_v, that an infinite trajectory will go through v equals the product of the edge-labels on the root-to-v path. In visualizing the process tree it will be helpful to draw each vertex v at Euclidean distance $-\log_2 p_v$ from the root. This way all trajectories whose last vertex is at the same distance from the root are equiprobable, even though they may entail wildly different numbers of steps (this also means that sibling vertices are not necessarily equidistant from the root). Finally, we color the vertices of the process tree as follows. For every infinite path that starts at the root determine its maximal prefix forming a bad trajectory. Color the vertices of the prefix red and the remaining vertices of the path blue.

In terms of the above picture, our goal will be to prove that there exist a critical radius x_0 and $\delta > 0$, such that the proportion of red states at distance x_0 from the root is at most $1 - \delta$. Crucially, x_0 will be polynomial, in fact linear, in $m = |F|$ and $\log_2 |\Omega|$. Since we will prove this for every possible initial state and the process is Markovian, it follows that the probability that the process reaches distance x from the root while going only through red states is at most $(1 - \delta)^{\lfloor x/x_0 \rfloor}$.

To prove that red vertices thin out as we move away from the root we stratify the process tree as follows. Fix any real number $x > 0$ and on each infinite path from the root mark the first vertex of probability 2^{-x} or less, i.e., the first vertex that has distance at least x from the root. Truncate the process tree so that the marked vertices become leaves of a finite tree. Let $L(x)$ be the set of all root-to-leaf paths (trajectories) in this finite tree and let $B(x) \subseteq L(x)$ consist of the bad trajectories. Now, let I be the random variable equal to an infinite trajectory of the process and let $\Sigma = \Sigma(x)$ be the random variable equal to the prefix of I that lies in $L(x)$. By definition, $\sum_{\ell \in L(x)} \Pr[\Sigma = \ell] = 1$, while $\Pr[\ell] \in (2^{-x-B}, 2^{-x}]$ for every $\ell \in L(x)$, since $-\log_2 \rho \geq B$. Let $P = P(\Sigma)$ be the maximal red prefix of Σ and observe that if $\Sigma \in B(x)$ then $P = \Sigma$. Therefore,

$$H[P] \;\geq\; \sum_{\ell \in B(x)} \Pr[\Sigma = \ell](-\log_2 \Pr[\Sigma = \ell]) \;\geq\; x \sum_{\ell \in B(x)} \Pr[\Sigma = \ell] \;=\; x \Pr[\Sigma \in B(x)] \;. \quad (5)$$

Assume now that there exist $M_0 > 0$ and $\lambda < 1$, such that $H[P] \leq \lambda x + M_0$, for every $x > 0$. Then (5) implies that for $x_0 = 2M_0/(1 - \lambda)$,

$$\Pr[\Sigma \in B(x_0)] \leq \frac{H[P]}{x_0} \leq \frac{\lambda x_0 + M_0}{x_0} = \lambda + \frac{1 - \lambda}{2} = \frac{1 + \lambda}{2} < 1 \;. \quad (6)$$

If $\Sigma \in B(x_0)$, we treat the reached state as the root of a new finite tree and repeat the same analysis, as it is independent of the starting state. It follows in this manner that for every integer $T \geq 1$, the probability that the process reaches a state at distance $T(x_0 + B)$

or more from the root by going only through red states is at most $((1+\lambda)/2)^T$. Thus, for any $s > 1/2$, the probability that the process reaches a state at distance

$$E = \left\lceil \frac{2s}{1+\lambda} \right\rceil (x_0 + B) = O\left(\frac{sM_0}{1-\lambda^2}\right)$$

or more from the root by going only through red states is at most $((1+\lambda)/2)^{\left\lceil \frac{2s}{1+\lambda} \right\rceil} < \exp(-s)$.

Since $\rho(\sigma,\tau) < 1-2^{-B}$, it follows that $-\log_2 \rho(\sigma,\tau) > 2^{-B}$, for every arc in D. Thus, after $2^B E$ steps the process is always at distance E or more from the root. Thus, the probability of not reaching a flawless state after $2^B E = O\left(\frac{sM_0}{1-\lambda^2}\right)$ steps is $\exp(-s)$. Therefore Theorem 1 follows from the following.

▶ **Theorem 3.** *Let* $\Xi = \max\{\mathrm{b_{ns}}, \mathrm{b_{po}}\}$ *and* $\Delta = \max_{j \in F} \Delta_j$. *If there exists* $\lambda < 1$ *such that for all* $j \in [m]$,

$$\sum_{f_i \in \Gamma_{\mathrm{po}}(f_j)} 2^{-(\lambda \mathrm{Potential}(f_i) - \mathrm{b}_{\mathrm{po}}^{f_i} - q_i(p))} < 2^{-(2+h(p))} \ ,$$

then $H[P] \le \lambda x + M_0$ *for every* $x > 0$, *where* $M_0 = \log_2 |\Omega| + m(\Delta+1)(\Xi+4) + \lambda B$.

The proof of Theorem 3 can be found in the full version of the paper [1]. To bound the entropy $H[P]$ we show how to represent trajectories as break sequences, described in the next section, and then show how to bound the entropy of break sequences by showing that they can be compressed in fewer bits, on average, than those consumed by the algorithm.

5 Break Sequences

Recall that π is an arbitrary but fixed ordering of the set of flaws F and that the highest flaw present in each state σ is denoted by $\pi(\sigma)$. We will refer to $\pi(\sigma)$ as the flaw *addressed* at state σ, i.e., as in the noiseless case, even though the action distribution $P(O(\sigma))$ may be "misguided" whenever $O(\sigma) \neq U(\sigma)$.

▶ **Definition 4.** Given a bad t-trajectory Σ, its *witness* sequence is $W(\Sigma) = w_1, \ldots, w_t = \{\pi(\sigma_i)\}_{i=1}^t$.

To prove Theorem 3, i.e., to gain control of bad trajectories and thus of $H[P]$, we introduce the notion of *break sequences* (see also [3, 2]). Recall that $U(\sigma)$ denotes the set of flaws present in σ.

▶ **Definition 5.** Let $B_0 = U(\sigma_1)$. For $1 \le i \le t-1$, let $B_i = U(\sigma_{i+1}) \setminus (U(\sigma_i) \setminus w_i)$.

Thus, B_i is the set of flaws "introduced" during the i-th step, where if a flaw is addressed in a step but remains present in the resulting state we say that it "introduced itself". Each flaw $f \in B_i$ may or may not be addressed during the rest of the trajectory. For example, f may get fixed "collaterally" during some step taken to address some other flaw, before the controller had a chance to address it. Alternatively, it may be that f remains present throughout the rest of the trajectory, but in each step $i < j \le t-1$ some other flaw has greater priority than f. It will be crucial to identify and focus on the subset of flaws $B_i^* \subseteq B_i$ that *do* get addressed during the t-trajectory, causing entropy to enter the system. Per the formal Definition 6 below, the set of such flaws is $B_i^* = B_i \setminus \{O_i \cup N_i\}$, where O_i comprises any flaws in B_i that get eradicated collaterally, while N_i comprises any flaws in B_i that remain present in every subsequent state after their introduction without being addressed.

▶ **Definition 6.** The *Break Sequence* of a t-trajectory is $B_0^*, B_1^*, \ldots, B_t^*$, where for $0 \leq i \leq t$,

$$B_i^* = B_i \setminus \{O_i \cup N_i\} \text{ , where}$$
$$O_i = \{f \in B_i \mid \exists j \in [i+1, t] : f \notin U(\sigma_{j+1}) \wedge \forall \ell \in [i+1, j] : f \neq w_\ell\} \ ,$$
$$N_i = \{f \in B_i \mid \forall j \in [i+1, t] : f \in U(\sigma_{j+1}) \wedge \forall \ell \in [i+1, t] : f \neq w_\ell\} \ .$$

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

$$E_{i+1} = (E_i - w_i) \cup B_i^* \ . \tag{7}$$

Observe that, by construction, $E_i \subseteq U(\sigma_i)$ and $w_i \in E_i$. Therefore, for every i, the highest flaw in E_i is w_i.

References

1 D. Achlioptas, F. Iliopoulos, and N. Vlassis. Stochastic Control via Entropy Compression. *ArXiv e-prints*, July 2016. URL: `https://arxiv.org/abs/1607.06494`, `arXiv:1607.06494`.

2 Dimitris Achlioptas and Fotis Iliopoulos. Focused stochastic local search and the Lovász local lemma. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 2024–2038. SIAM, 2016. `doi:10.1137/1.9781611974331.ch141`.

3 Dimitris Achlioptas and Fotis Iliopoulos. Random walks that find perfect objects and the Lovász local lemma. *J. ACM*, 63(3):22, 2016. `doi:10.1145/2818352`.

4 Mikko Alava, John Ardelius, Erik Aurell, Petteri Kaski, Supriya Krishnamurthy, Pekka Orponen, and Sakari Seitz. Circumspect descent prevails in solving random constraint satisfaction problems. *Proceedings of the National Academy of Sciences*, 105(40):15253–15257, 2008. `doi:10.1073/pnas.0712263105`.

5 Noga Alon. A parallel algorithmic version of the local lemma. *Random Struct. Algorithms*, 2(4):367–378, 1991. `doi:10.1002/rsa.3240020403`.

6 József Beck. An algorithmic approach to the Lovász local lemma. I. *Random Structures Algorithms*, 2(4):343–365, 1991. `doi:10.1002/rsa.3240020402`.

7 Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. II*. Athena Scientific, 2012.

8 Krishnendu Chatterjee, Martin Chmelik, Raghav Gupta, and Ayush Kanodia. Optimal cost almost-sure reachability in POMDPs. *Artif. Intell.*, 234:26–48, 2016. `doi:10.1016/j.artint.2016.01.007`.

9 Lonnie Chrisman. Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. In *AAAI*, pages 183–188. Citeseer, 1992.

10 Artur Czumaj and Christian Scheideler. Coloring non-uniform hypergraphs: a new algorithmic approach to the general Lovász local lemma. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms (San Francisco, CA, 2000)*, pages 30–39, 2000.

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

12 Kousha Etessami and Mihalis Yannakakis. On the complexity of Nash equilibria and other fixed points. *SIAM Journal on Computing*, 39(6):2531–2597, 2010.

13 Jerzy Filar and Koos Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag New York, Inc., New York, NY, USA, 1996.

14 David G. Harris and Aravind Srinivasan. A constructive algorithm for the Lovász local lemma on permutations. In *SODA*, pages 907–925. SIAM, 2014. doi:10.1137/1.9781611973402.68.

15 Nicholas J. A. Harvey and Jan Vondrák. An algorithmic proof of the Lovász local lemma via resampling oracles. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1327–1346. IEEE Computer Society, 2015. URL: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7352273, doi:10.1109/FOCS.2015.85.

16 Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

17 Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

18 Kashyap Babu Rao Kolipaka and Mario Szegedy. Moser and Tardos meet Lovász. In *STOC*, pages 235–244. ACM, 2011. doi:10.1145/1993636.1993669.

19 Vladimir Kolmogorov. Commutativity in the algorithmic lovász local lemma. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 780–787. IEEE Computer Society, 2016. URL: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7781469, doi:10.1109/FOCS.2016.88.

20 Michael L. Littman, Judy Goldsmith, and Martin Mundhenk. The computational complexity of probabilistic planning. *Journal of Artificial Intelligence Research*, 9(1):1–36, 1998.

21 Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In *AAAI/IAAI*, pages 541–548, 1999.

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

23 Robin A. Moser. A constructive proof of the Lovász local lemma. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 – June 2, 2009*, pages 343–350. ACM, 2009. doi:10.1145/1536414.1536462.

24 Robin A. Moser and Gábor Tardos. A constructive proof of the general Lovász local lemma. *J. ACM*, 57(2):Art. 11, 15, 2010. doi:10.1145/1667053.1667060.

25 Martin Mundhenk, Judy Goldsmith, Christopher Lusena, and Eric Allender. Complexity of finite-horizon Markov decision process problems. *Journal of the ACM (JACM)*, 47(4):681–720, 2000.

26 Christos H. Papadimitriou. On selecting a satisfying truth assignment (extended abstract). In *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1-4 October 1991*, pages 163–169. IEEE Computer Society, 1991. doi:10.1109/SFCS.1991.185365.

27 Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

28 Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

29 Bart Selman, Henry A. Kautz, and Bram Cohen. Local search strategies for satisfiability testing. In David S. Johnson and Michael A. Trick, editors, *Cliques, Coloring, and Satisfiability, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, October 11-13, 1993*, volume 26 of *DIMACS Series in Discrete Mathematics*

and Theoretical Computer Science, pages 521–532. DIMACS/AMS, 1993. URL: `http://dimacs.rutgers.edu/Volumes/Vol26.html`.

30 Aravind Srinivasan. Improved algorithmic versions of the Lovász local lemma. In Shang-Hua Teng, editor, *SODA*, pages 611–620. SIAM, 2008. URL: `http://dl.acm.org/citation.cfm?id=1347082.1347150`.

31 Nikos Vlassis, Michael L. Littman, and David Barber. On the computational complexity of stochastic controller optimization in POMDPs. *ACM Transactions on Computation Theory (TOCT)*, 4(4):12, 2012. `doi:10.1145/2382559.2382563`.