# Resilient Strategies for Stochastic Systems: How Much Does It Take to Break a Winning Strategy?

AAAI Track

Kush Grover
Fondazione Bruno Kessler & Masaryk University
Trento, Italy

Markel Zubia
Ruhr University Bochum
Bochum, Germany

Debraj Chakraborty
Nanyang Technological University & Masaryk University
Singapore, Singapore

Muqsit Azeem
Technical University of Munich & University of Konstanz
Munich, Germany

Nils Jansen
Ruhr University Bochum & Radboud University Nijmegen
Bochum, Germany

Jan Křetínský
Masaryk University & Technical University of Munich
Brno, Czech Republic

## ABSTRACT

We study the problem of resilient strategies in the presence of uncertainty. Resilient strategies enable an agent to make decisions that are robust against disturbances. In particular, we are interested in those disturbances that are able to flip a decision made by the agent. Such a disturbance may, for instance, occur when the intended action of the agent cannot be executed due to a malfunction of an actuator in the environment. In this work, we introduce the concept of resilience in the stochastic setting and present a comprehensive set of fundamental problems. Specifically, we address these problems for Markov decision processes with reachability and safety objectives, which also smoothly extend to stochastic games. We provide various ways of aggregating the amounts of disturbances that may have occurred, for instance, in expectation or in the worst case. Moreover, to reason about infinite disturbances, we use quantitative measures, like their frequency of occurrence.

## KEYWORDS

Markov Decision Processes; Stochastic Games; Verification

**ACM Reference Format:**
Kush Grover, Markel Zubia, Debraj Chakraborty, Muqsit Azeem, Nils Jansen, and Jan Křetínský. 2026. Resilient Strategies for Stochastic Systems: How Much Does It Take to Break a Winning Strategy?: AAAI Track. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/JAGQ1479

## 1 INTRODUCTION AND MOTIVATION

In many areas such as machine learning, robotics, automated planning, and game theory, a notion of optimality is adopted to argue about the best possible behavior in a given environment. While optimality captures the best possible performance an agent may achieve in a fixed, well-defined environment, real-world settings are rarely stable or predictable. Then, resilience becomes a more meaningful objective: it reflects the agent's ability to maintain desirable behavior despite changes or adversarial disturbances. The latter includes notions such as fault tolerance of algorithms, robustness to disturbances in control, or trembling hands equilibria in game theory. As Vardi [27] argues, this notion is spread not only over computer science, but also economics, evolution, or other dynamic systems; and while considerable effort has been spent in the area, still *"We must recognize the trade-off between efficiency and resilience. It is time to develop the discipline of resilient algorithms."*

*Resilient Strategies.* This paper focuses on resilience of *strategies* (a.k.a. policies, schedulers, or controllers, depending on the context) in terms of the number of decisions that need to be subverted, which we refer to as *disturbances*, to violate the property otherwise satisfied by the original strategy. From the control perspective, such disturbances may represent actuator faults, control noise, or environmental perturbations that prevent the agent from executing its intended action. For instance, consider an autonomous drone navigating a grid world toward a target (Figure 1): the strategy corresponding to the red path would reach the target in the ideal situation. However, it passes close to the trees, making it vulnerable to collisions from disturbances (due to wind). A resilient strategy, in contrast, would follow the green path which maintains a buffer from the trees. This leads to the target with probability 1, but it can also handle a disturbance. Understanding how sensitive a strategy is to such disturbances is essential for deploying agents in uncertain or adversarial environments.

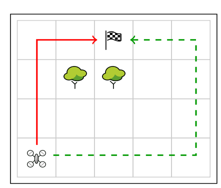

**Figure 1: Drone navigation task with wind disturbances.**

From a theoretical perspective, analyzing resilience reveals which regions of the state space are critical to maintaining correct behavior. That is, it helps identify the brittle points of the agent's strategy where a few disturbances can lead to failure. This information is



*Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus.* © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/JAGQ1479

relevant in the context of *explaining* its key decisions [3], in the context of *strategy repair* [22], and in the modern on-the-fly automated synthesis approaches [15]. Notably, standard modeling approaches such as incorporating noise probabilistically or assuming a fully adversarial environment fall short of capturing this nuance. Probabilistic models mask the fragility of decisions by averaging over expected behaviors, while adversarial models are overly pessimistic and can rule out efficient paths unnecessarily. In contrast, our disturbance-based view offers a finer robustness metric by asking how many deviations are needed to cause failure.

*Resilience in Stochastic Systems.* Previous work has investigated the resilience of strategies in (non-quantitative) graph games [20]. While the landscape is rather bland there, it becomes very vibrant in the stochastic context. Indeed, in the former setting, (i) quantifying resilience by the number of disturbances (decisions to be subverted) during a play is a straightforward choice, (ii) it results in an integer bounded by the size of the state space or infinity, (iii) algorithmically, it boils down to iterative, graph-search-based procedures that identify regions from which an agent can enforce reaching a goal regardless of disturbances. In stark contrast, for Markov decision processes and stochastic games, (i) resilience purely in terms of expected disturbance counts may not be adequate, and we also analyze worst cases happening with positive probability; (ii) the number of disturbances may now be larger than the state space size and is, in fact, unbounded; in such cases, we refine the resilience measure by considering the frequency of disturbances rather than their total count; (iii) algorithmically, only some cases can be easily reduced to simple graph search or mean-payoff computations.

*Assumptions on Model Availability.* Our work relies on the standard assumption in the model-based planning paradigm that the underlying model is available in explicit form or can be obtained through established learning methods. This convention is well documented in the model-checking literature; see, for example [5], where verification algorithms are defined relative to a given system model. When the model is not provided a priori but is learnable from data, statistical model checking offers a widely used pathway for reconstructing or approximating the relevant stochastic behavior [1, 4]. Once such an approximation is obtained, the resulting model fits within the standard verification framework and our method applies directly.

*Disturbances with Different Costs.* In this work, we focus on symmetric disturbances, i.e. all disturbances have the same cost, for two reasons: (i) Prior work on resilience in non-stochastic settings [11, 20] has considered only symmetric disturbances, and we aim to extend that line of work to stochastic settings. (ii) From the practical perspective, assigning precise costs to disturbances is often challenging, as it requires detailed domain knowledge. On the other hand, symmetric disturbances appropriately model several realistic scenarios, for example network routing problems [14, 23]. Furthermore, our framework can naturally extend to asymmetric disturbance costs by assigning different weights to disturbance transitions. This would change the resilience metric from "how many disturbances" to "total disturbance cost". This would require adapting some algorithms presented here, while the rest would remain unchanged, and we leave this for future work.

*Summary of Our Contribution:*

- We extend the notion of strategy resilience to the stochastic setting with safety and reachability objectives. To that end, we consider expected and positive-measure worst cases. When infinitely many disturbances are required to break a strategy, we refine the notion by computing their frequency.
- We provide algorithms for computing the resilience of a given strategy (for each definition of resilience introduced) and to compute optimally resilient strategies. From the perspective of the efficiency-resilience trade-off, these are satisfying strategies (satisfy the functional property with at least a given probability threshold) with optimal resilience (requiring the most disturbances to break them, i.e., decrease the probability below the threshold).

The resilience of a strategy is measured by calculating its *breaking point*, defined as the maximum number of disturbances that are necessary to break it. Table 1 provides an overview of the algorithms to compute the breaking point of a pure memoryless strategy in the worst and expected cases, and for finding a strategy with the maximum breaking point. Definitions of terms used in the tables are provided in Section 2. Due to space limitations, proofs and other details are provided in the appendix of the full version [13].

*Related Work.* The concept of system robustness against errors has been extensively studied in various contexts. One common approach is to model uncertainties in estimated probabilities using an uncertainty set within which the true probability resides. To ensure absolute safety, a worst-case analysis is often employed, yielding results that are resilient to such disturbances [7]. Another approach is to perform a sensitivity analysis over the uncertainty sets to measure how resilient the system is with respect to different variables [29]. For a comprehensive comparison of different notions of resilience and robustness, we refer the reader to [11, 20].

Apart from uncertainty sets, notions of differential privacy and deception for MDPs have also been explored, which utilize and optimize for different measures of detection and resilience [6, 12, 17]. Another way to model disturbances is by treating them as random events with small probabilities, often called *trembling hand* [18, 28]. However, this might not always be suitable, and defining an accurate stochastic error model may be challenging, also argued in [20].

Therefore, the idea of strategies that are resilient to unmodeled intermittent disturbances was first introduced in [11] within the context of safety games. This concept was later extended to prefix-independent winning conditions, including parity objectives, in [20]. It was demonstrated that computing optimally resilient strategies for parity conditions incurs only a polynomial overhead compared to solving traditional parity games. These ideas were further applied in [25] to find resilient controllers for a continuous dynamical system. Additionally, the notion of resilient strategies was extended to infinite arenas, particularly pushdown graphs, in [19].

However, none of these works consider the potential benefits of leveraging partial knowledge of stochasticity within the model. To our knowledge, this work is the first to study unmodeled intermittent disturbances in the context of stochastic systems such as Markov decision processes and stochastic games. Moreover, prior works do not investigate resilience in terms of the frequency of

**Table 1: Overview of strategy evaluation and optimal strategy computation under different semantics and breaking point types.**

| Semantics | Breaking Point Type | Strategy Evaluation | | Strategy Synthesis | |
|---|---|---|---|---|---|
| | | Safety | Reachability | Safety | Reachability |
| Expected | Transient | SSP for MDP **(P)** | SSP for MDP **(P)** | SSP for SG | SSP for SG |
| | Frequency | Cannot happen | Collapse MECs + SSP for MDP **(P)** | Cannot happen | Collapse MECs + SSP for SG |
| Worst-case | Transient | Iterative LP **(PSPACE)** | Iterative LP **(PSPACE)** | Iterative QP **(PSPACE)** | Iterative QP **(PSPACE)** |
| | Frequency | Always 0 | Collapse MECs + Worst-case analysis **(P)** | Always 0 | Collapse MECs + Worst-case analysis **(NP)** |

disturbances required to compromise a strategy, particularly in scenarios where infinite disturbances can be handled by the controller.

## 2 PRELIMINARIES

For a set $A$, we denote its power set by $Pow(A)$. We use Von Neumann ordinals, with $\langle n \rangle = \{0, \ldots, n-1\}$ for all $n \in \mathbb{N}$. The set of all (discrete) probability distributions on the set $S$ is denoted by $\mathrm{Dist}(S)$. We denote by $S^*$ the set of all *finite strings* over $S$.

### 2.1 Stochastic Games

Definition 1. *A (2-player)* Stochastic Game *(SG) is a tuple* $\mathcal{G} = (S, S_1, S_2, A, Av, T)$ *where $S$ is a finite set of states partitioned into Player 1 states $S_1$ and Player 2 states $S_2$; $A$ is a finite set of actions; $Av : S \to Pow(A)$ is a total function which maps the set of available actions to each state, and $T : S \times A \rightharpoonup \mathrm{Dist}(S)$ is a partial transition function, which, given a state and an available action, returns a probability distribution over the successor states.*

We also define an initial state, $s_0 \in S$ from which the play starts. With a slight abuse of notation, we use $T(s, a, s')$ to denote $T(s, a)(s')$. For this paper, we assume that the stochastic game is turn-based, i.e., in each state, only one player can move.

*Paths or Runs.* The set of successor states for $s \in S$ that can be reached by taking the action $a$ is $\mathrm{Post}^{\mathcal{G}}(s, a) = \{s' \mid T(s, a, s') > 0\}$. A *finite path* (or *finite run*) $\varrho = s_0 a_0 s_1 \ldots s_i$ of length $i \geq 0$ is a sequence of states and actions such that for all $t \in [0, i-1]$, $a_t \in Av(s_t)$ and $s_{t+1} \in \mathrm{Post}^{\mathcal{G}}(s_t, a_t)$. We can define infinite paths (or infinite runs) $\rho = s_0 a_0 s_1 a_1 s_2 \ldots$ analogously.

*Strategies.* A Player 1 strategy is defined as a function $\pi : (S \times A)^* \times S_1 \to \mathrm{Dist}(A)$ mapping the history to a distribution over available actions. A strategy is *pure* if the distribution is always a Dirac delta function, and it is *mixed* otherwise. Player 2 strategies have analogous definitions.

*Markov Decision Processes (MDP) and Markov Chains (MC).* MDPs are a special case of stochastic games where the states of one player are empty i.e. $S_2 = \emptyset$. If the strategy of Player 1, $\pi$, is fixed in the SG $\mathcal{G}$, it induces an MDP $\mathcal{G}^{\pi}$. An MC is an MDP with $|Av(s)| = 1$ for all $s \in S$. Fixing a strategy $\sigma$ on an MDP $\mathcal{G}^{\pi}$ induces a Markov chain $\mathcal{G}^{\pi,\sigma}$. An MC $M$ and an initial state $s_0$ define a unique probability measure $\mathbb{P}_{M,s_0}$ over infinite paths [24]. For any random variable $X$ defined over the infinite paths of MC $M$, its expected value with respect to $\mathbb{P}_{M,s_0}$ is $\mathbb{E}_{M,s_0}[X]$.

*End Component (EC).* A set of states and actions is an *end component* if the play never leaves that set and it is a *maximal end component* (MEC) if it cannot be extended by adding more states and actions. The states in an MDP can be partitioned into MECs, and every play is guaranteed to eventually enter a MEC. We refer to standard literature [24] for the formal definitions. We use $\mathsf{MEC}(M)$ to denote the set of maximal end components of $M$. This notion of MECs can be extended to SGs.

*Rewards or Costs.* We model the costs associated with each state-action pair using a function $C : S \times A \to \mathbb{N}$. Given a Markov chain (MC) $M$, let $C_i$ be a random variable that, for an infinite path $\rho = s_0 a_0 s_1 a_1 \ldots$, returns $C_i(\rho) = C(s_i, a_i)$; i.e., the cost (or reward) incurred at the $i$-th step of the path. The total cost for an infinite path $\rho$ is defined as $C_\rho := \sum_{j=0}^{\infty} C(s_j, a_j)$. The *expected total cost* for $M$ starting from state $s_0$ is defined as $\mathsf{TR}(M, s_0, C) := \sum_{k=0}^{\infty} \mathbb{E}_{M,s_0}(C_k) = \mathbb{E}_{M,s_0}(C_\rho)$. The $k$-step average cost from state $s_0$ in $M$ is defined as $v_k(s_0) := \mathbb{E}_{M,s_0}\left(\frac{1}{k+1}\sum_{j=0}^{k} C_j\right)$. The *expected mean payoff* for $M$ from state $s_0$ is defined as $\mathsf{MP}(M, s_0, C) := \liminf_{k\to\infty} v_k(s_0)$. The mean payoff for a path $\rho$ is defined as $\mathsf{MP}(\rho) := \liminf_{k\to\infty} \frac{1}{k+1}\sum_{j=0}^{k} C(s_j, a_j)$.

*Specifications.* We use the standard temporal operators $\Box$ (globally) and $\Diamond$ (eventually). In an MC $M$ with initial state $s_0$, we let $\mathbb{P}_{\mathcal{G},s_0}(\Box\neg B)$ denote the probability of *never* visiting $B \subseteq S$, and $\mathbb{P}_{\mathcal{G},s_0}(\Diamond G)$ denote the probability of *eventually* reaching $G \subseteq S$. We focus on two standard quantitative specifications and their negations: *safety*, where $\phi_{safety} := \mathbb{P}_{\mathcal{G},s_0}(\Box\neg B) > p$; and *reachability*, where $\phi_{reach} := \mathbb{P}_{\mathcal{G},s_0}(\Diamond G) > q$. The properties $\phi_{safety}$ and $\phi_{reach}$ could equivalently be stated with $\geq$; the analysis carries over with only minor adaptations. Under expected semantics, the same computation yields *resilience* rather than the *breaking point*, while the worst-case semantics remains unchanged. W.l.o.g., we assume that all states in $G$ or $B$ are sink states with only a self loop.

*Algorithms to Solve MDPs and SGs.* For solving *reachability* and *safety* problems in MDPs and SGs, key algorithms includes Value Iteration (VI), Linear (LP) and Quadratic Programming (QP), and Policy Iteration (PI). We refer the reader to standard texts [9, 24] for details of these algorithms. For solving *stochastic shortest path* (SSP) problems, that ask to minimize the cost of reaching a target state, one can use LP for MDPs and VI/PI for SGs.

## 3 STOCHASTIC GAMES WITH DISTURBANCES

In this section, we define stochastic games with disturbances. In these systems, disturbances may occur at runtime and override the decisions of Player 1. For example, as seen in Fig. 1, if the drone wants to move upwards, wind can disturb its actions and push it to the right, completely changing its decision.

Definition 2. *A* Stochastic Game with Disturbances *(SGD) is a tuple* $\mathcal{G} = (S, S_1, S_2, A, Av, T, A^D, Av^D, T^D)$ *where* $(S, S_1, S_2, A, Av, T)$ *is an SG,* $A^D$ *is a finite set of* disturbance actions *(disjoint from A),* $Av^D : S_1 \to Pow(A^D)$ *is a function specifying the* available disturbance actions *in Player 1 states, and* $T^D : S_1 \times A^D \rightharpoonup \mathrm{Dist}(S)$ *defines a* disturbance transition function.

We use $|T^D|$ to denote the number of disturbance transitions in the game. We can define an initial state and strategies for both players in the same way as for an SG. However, since we have disturbance actions here, we also define a disturbance strategy. A *disturbance strategy* is defined as $\delta : (S \times A)^* \times S_1 \to Dist(A^D \cup \{\bot\})$. Here, $\bot$ represents that no disturbance action is taken. A run of the game is a sequence of states and actions $\rho = s_0 a_0 s_1 a_1 \dots$ where $\forall i, s_i \in S,\ a_i \in Av(s_i) \cup Av^D(s_i)$, and $s_{i+1} \in T(s_i, a_i)$. For a run $\rho = s_0 a_0 s_1 a_1, \dots$, we define the total number of disturbances as $\mathcal{D}^T(\rho) := \big|\{a_i \mid a_i \in Av^D(s_i)\}\big|$. This number may not be finite, and in that case, we can define the frequency of disturbances as $\mathcal{D}^F(\rho) := \liminf_{k\to\infty} \frac{1}{k}\big|\mathcal{D}^T(\rho[0,k])\big|$, where $\rho[0,k]$ denotes the $k$-length prefix of an infinite run $\rho$. We need lim inf here because the limit in general might not exist.

*Modeling Disturbances as Costs.* We can model the disturbances as costs in an SGD by letting $C(s,a) = 1$ if $s \in S_1, a \in A^D$, and letting $C(s,a) = 0$ otherwise.

**Remark 1.** *Note that for a run* $\rho$*, the number of disturbances and the frequency of disturbances can be seen as the total cost and the mean payoff of the run respectively, i.e.*

$$\mathcal{D}^T(\rho) = C_\rho \text{ and } \mathcal{D}^F(\rho) = \mathsf{MP}(\rho)\,.$$

## 3.1 Resilience, Breaking Point, and Their Different Semantics

Since disturbances are adversarial, we can assume that they are performed by Player 2. The *resilience* of a strategy is defined as the maximum number of disturbances that the strategy can withstand while satisfying a given specification. However, this maximum might not exist, for instance if it can withstand < 2.5 disturbances on average but cannot withstand 2.5 disturbances. Therefore, a more suitable, well-defined notion would be the *breaking point* of a strategy, which refers to the minimum number of disturbances required to break it. In this setting, *an* ***optimally resilient strategy*** *is a strategy that has the greatest breaking point.*

There are various natural ways to quantify how many disturbances are required to break a strategy. One approach is to consider the expected number of disturbances needed to break said strategy. Another approach is to determine the maximum number of disturbances required over all possible paths, which we refer to as the worst-case measure throughout this paper. The choice between these measures depends on the context. Expected resilience captures average-case behavior and is appropriate when disturbances occur frequently. However, it may obscure rare but critical scenarios in which only a few disturbances suffice to break the strategy. For robustness guarantees, it is often more informative to know how many disturbances the strategy can withstand in the worst case.

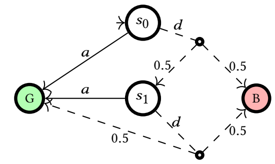


**Figure 2: For this SGD, if the objective is to reach $G$ with probability $> 0.4$, then the worst-case breaking point is 2, as disturbing just once will not break the policy, and the expected breaking point is 1.1 via an adversary that always disturbs in $s_0$ and with a probability of 0.2 in $s_1$.**

## 3.2 Induced MDP Under a Player 1 Strategy

Given an SGD $\mathcal{G}$ and a memoryless Player 1 strategy $\pi$, we use $M_\pi$ to denote the *induced MDP* under $\pi$, described next. For every state $s \in S_1$, we remove all Player 1 actions except for the one picked by $\pi(s)$, while the disturbance actions remain unchanged. Now, we can think of all states as Player 2 states.

**Lemma 1.** *For a memoryless Player 1 strategy* $\pi$*, there exist a bijective function h that maps pairs* $(\sigma, \delta)$ *of Player 2 and disturbance strategies in* $\mathcal{G}$ *to a strategy* $\mu$ *in* $M_\pi$ *such that*

$$\forall \rho \in (S \times A)^* \times S, \quad \mathbb{P}^{\pi,\sigma,\delta}_{\mathcal{G},s_0}(\rho) = \mathbb{P}^{\mu}_{M_\pi,s_0}(\rho).$$

This lemma implies that the pair of strategies $(\sigma, \delta)$ for $\mathcal{G}$ is equivalent to strategy $\mu$ for MDP $M_\pi$.

For a Player 1 strategy $\pi$, a Player 2 strategy $\sigma$, a disturbance strategy $\delta$ and an initial state $s_0$, we can define the induced MC as follows: In Player 2 states, the action is selected by $\sigma$. In Player 1 states, if $\delta(s) = \bot$ then we pick action $\pi(s)$, whereas we pick $\delta(s)$ otherwise. We denote the probability measure in the induced MC as $\mathbb{P}^{\pi,\sigma,\delta}_{\mathcal{G},s_0}$. For any random variable $X$ defined over infinite paths in $\mathcal{G}$, its expected value with respect to $\mathbb{P}^{\pi,\sigma,\delta}_{\mathcal{G},s_0}$ is $\mathbb{E}^{\pi,\sigma,\delta}_{\mathcal{G},s_0}(X)$.

## 3.3 From SGDs with Finite Disturbances to SGs

Here, we show that if the number of disturbances is bounded (by some $k \in \mathbb{N}$) on all runs, then we can transform an SGD $\mathcal{G}$ into an equivalent SG $\mathcal{G}^{\dagger k}$. This is achieved by unfolding the state space of the SGD to account for the number of disturbances that have already occurred. Specifically, we encode the remaining number of disturbances into the state space, i.e., the new states are of the form $(s, i)$, where $s$ is a state in $\mathcal{G}$ and $i \in \langle k+1 \rangle$ represents the number of remaining disturbances. Fig. 3 illustrates this transformation.

Intuitively, after Player 1 chooses an action in a state $s \in S_1$, a disturbance may or may not occur. To embed this extra step into a standard stochastic game, we create an intermediate state $(s, i, a)$, controlled by Player 2, that is reached whenever Player 1 plays action $a$ from $s$. Player 2 then chooses either $\bot$ (no disturbance) or a disturbance action $d$. If $\bot$ is chosen, we proceed with the usual transition; if $d$ is chosen, the transition follows $T^D$ and we decrement the disturbance counter $i$. Hence, each disturbance possibility becomes a normal turn-based move of Player 2, making the disturbance mechanism explicit in the resulting unfolded game. It can easily be shown that this transformation is sound by showing the equivalence of the two games.

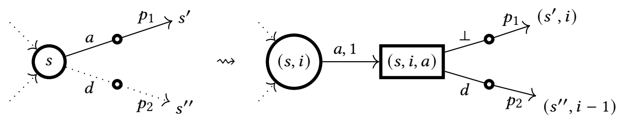


**Figure 3: A gadget in the unfolded stochastic game.**

**Lemma 2.** *There exist bijective functions $f$ and $g$ where $f$ maps triplets $(\pi, \sigma, \delta)$ of strategies in $\mathcal{G}$ to pairs $(\tilde{\pi}, \tilde{\sigma})$ of strategies in $\mathcal{G}^{\dagger k}$, and $g$ maps paths $\rho$ in $\mathcal{G}$ to paths $\tilde{\rho}$ in $\mathcal{G}^{\dagger k}$, such that $\forall \rho \in (S \times A)^* \times S$, $\mathbb{P}^{\pi,\sigma,\delta}_{\mathcal{G},s_0}(\rho) = \mathbb{P}^{\tilde{\pi},\tilde{\sigma}}_{\mathcal{G}^{\dagger k},(s_0,k)}(\tilde{\rho})$, as long as $\delta$ disturbs at most $k$ times on each run.*

## 4 EXPECTED BREAKING POINT

For a given Player 1 strategy $\pi$, we define the set of pairs of strategies that break $\pi$ as $U_\pi := \{(\sigma, \delta) \mid \mathcal{G}^{\pi,\sigma,\delta}_{s_0} \models \neg\phi\}$. We define the *expected transient breaking point* of $\pi$ as

$$\mathcal{EB}^T_{\mathcal{G},\phi}(\pi) := \inf_{(\sigma,\delta)\in U_\pi} \mathbb{E}^{\pi,\sigma,\delta}_{\mathcal{G},s_0}\big(\mathcal{D}^T(\rho)\big).$$

When set $U_\pi$ is empty, which means that the strategy cannot be broken, we refer to it as $\natural$. When there exist no finite values in $U_\pi$, we refer to it as $\omega$.

If breaking the strategy requires infinitely many disturbances on average, we can still ask how frequently disturbances are required to break the strategy. For such cases, the *expected frequency breaking point* is defined as

$$\mathcal{EB}^F_{\mathcal{G},\phi}(\pi) := \inf_{(\sigma,\delta)\in U_\pi} \mathbb{E}^{\pi,\sigma,\delta}_{\mathcal{G},s_0}\big(\mathcal{D}^F(\rho)\big).$$

When $U_\pi$ is empty, we also denote it as $\natural$.

Definition 3. *Given an SGD $\mathcal{G}$ and an objective $\phi$, the* expected breaking point *of a Player 1 strategy $\pi$ is given by the pair*

$$\mathcal{EB}_{\mathcal{G},\phi}(\pi) := \big(\mathcal{EB}^T_{\mathcal{G},\phi}(\pi), \mathcal{EB}^F_{\mathcal{G},\phi}(\pi)\big).$$

Because of the observation in Remark 1 and Lemma 1, we get the following lemma.

**Lemma 3.** *Given an SGD, an objective $\phi$, and a memoryless Player 1 strategy $\pi$, it holds that $\mathcal{EB}^T_{\mathcal{G},\phi}(\pi) = \inf_{\mu\in U_\pi} \mathsf{TR}(M^\mu_\pi, s_0, C)$, and $\mathcal{EB}^F_{\mathcal{G},\phi}(\pi) = \inf_{\mu\in U_\pi} \mathsf{MP}(M^\mu_\pi, s_0, C)$.*

### 4.1 Computing the Expected Breaking Point

Next, we describe the algorithm for expected transient and frequency breaking points for both safety and reachability objectives. We compute the expected breaking point for memoryless strategies here[1]. We use the induced MDP $M_\pi$ and the cost function $C$ defined previously to compute it.

[1]It can easily be extended to finite memory strategies using standard techniques, e.g., encoding the memory in state space by taking product of the game with a finite automaton representing the strategy.

*Safety.* To violate the safety objective, we need to compute the minimum expected cost while reaching states in $B$ with probability $\geq 1 - p$. This is a variant of the standard SSP for which the solution can be found using the LP described in [16]. If there is no solution to the linear program, it is not possible to reach $B$ with the required probability, which implies that the expected breaking point does not exist. If a solution is found, the frequency breaking point is 0 and the transient breaking point is the solution to the LP. Note that the strategy generated by the LP can be mixed.

*Reachability.* To violate the reachability objective, we need to ensure the avoidance of states in the set $G$ with probability $\geq 1 - p$ while minimizing cost. We proceed as follows. Define $B$ to be the set of MECs in $M_\pi \setminus G$ where Player 2 can ensure to remain indefinitely without invoking any disturbances. These MECs can be identified by excluding any MEC of $M_\pi \setminus G$ that have exits via actions suggested by strategy $\pi$,

$$B := \{U \in \mathsf{MEC}(M_\pi \setminus G) \mid \forall s' \in U \cap S_1 : \mathsf{Post}^{M_\pi}(s', \pi(s')) \subseteq U\}. \quad (1)$$

We also define the set $R$ of MECs in which Player 2 can remain indefinitely but require disturbance actions to do so[2],

$$R := \{U \in \mathsf{MEC}(M_\pi \setminus G) \mid \forall s' \in U \cap S_1 : \mathsf{Post}^{M_\pi}(s', \pi(s')) \not\subseteq U \implies Av^D(s) \neq \emptyset\}. \quad (2)$$

The probabilities of reaching $B$ and $R \cup B$ can be calculated through standard model checking algorithms. Then, we identify 3 cases:

**Case 1: Probability to reach $B$ is $\geq 1 - p$.** The expected cost to reach $B$ is finite here, making the transient breaking point finite and the frequency breaking point 0. Therefore, the transient breaking point is computed using an SSP formulation with respect to the reachability probability.

**Case 2: Probability to reach $R \cup B$ is $< 1 - p$.** In this case, it is not possible to break the strategy as the goal states cannot be avoided with the required probability. Both the transient and frequency breaking points are $\natural$.

**Case 3: Probability to reach $B$ is $< 1 - p$ but $R \cup B$ is $\geq 1 - p$.** This is the intermediate case, where finitely many disturbances are not enough to break the strategy. Intuitively, Player 2 cannot force the play to reach states (with the required probability) from which no more disturbances are needed to break the Player 1 strategy. Therefore, the transient breaking point is $\omega$, and the expected frequency breaking point is computed as follows. For each MEC in $R$, the minimum expected mean payoff of staying inside the MEC is computed. This mean payoff represents the frequency of disturbances required to stay within the MEC. We then construct the weighted MEC quotient of $M_\pi$ (similar to [2]), by collapsing each MEC in $R$ to a single abstract state. A new cost function is defined for this quotient MDP. Each collapsed state is augmented with an outgoing transition to a fresh terminal state $s_+$, where the new cost of this transition corresponds to the mean payoff of the original MEC. The new cost assigned to all other transitions in the MDP is zero. Finally, we solve an SSP problem with the new cost on this modified MDP to compute the minimum expected cost of

[2]We slightly abuse notation by using $B$ and $R$ to also refer to the union of these MECs.

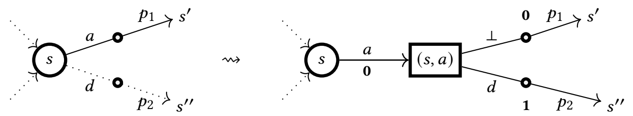


**Figure 4: Gadget for converting an SGD into an SG for the Expected Breaking Point.**

reaching $B \cup \{s_+\}$. The solution to this SSP is precisely the expected frequency breaking point of $\pi$.

**Theorem 1.** *Given an SGD $\mathcal{G}$, objective $\phi$, Player 1 strategy $\pi$, and pair of values $(t, f)$, deciding whether $\mathcal{EB}_{\mathcal{G},\phi}(\pi) \geq (t, f)$ is in* **P**.

## 4.2 Optimal Strategy: Expected Breaking Point

The optimally resilient strategy is defined as the strategy that achieves the breaking point

$$\mathcal{EB}_{\mathcal{G},\phi} = \big(\max_{\pi} \mathcal{EB}^{T}_{\mathcal{G},\phi}(\pi), \max_{\pi} \mathcal{EB}^{F}_{\mathcal{G},\phi}(\pi)\big).$$

To compute this, we construct a transformed stochastic game $\tilde{\mathcal{G}}$ from the given SGD $\mathcal{G}$ using the gadget in Fig. 4. This transformation follows the unfolded SG structure introduced in Section 3.3, but omits the use of duplicated copies. Specifically, for each state $s \in S_1$ and action $a \in Av(s)$, we introduce an intermediate Player 2 state $(s, a)$, reached deterministically when Player 1 selects action $a$ in state $s$. From $(s, a)$, Player 2 chooses either $\perp$, that follows the distribution $T(s, a)$ simulating the undisturbed execution of $a$, or a disturbance action $d$, which follows the distribution $T^D(s, d)$, simulating the effect of a disturbance. To capture the cost, disturbance actions are assigned cost 1, and all others cost 0, as indicated by the boldface labels in Fig. 4. The minimum cost incurred while violating $\phi$ in $\tilde{\mathcal{G}}$ corresponds to the breaking point. Thus, optimally resilient Player 1 strategies are generally memoryless and randomized for the expected breaking point. Optimal disturbance strategies are also memoryless and randomized; for instance, see Figure 2.

**Lemma 4.** *Given an SGD $\mathcal{G}$, objective $\phi$ and memoryless Player 1 strategy $\pi$, it holds that $\mathcal{EB}^{T}_{\mathcal{G},\phi}(\pi) = \inf_{\mu \in U_\pi} \mathsf{TR}(\tilde{\mathcal{G}}^{\pi,\mu}, s_0, C)$ and $\mathcal{EB}^{F}_{\mathcal{G},\phi}(\pi) = \inf_{\mu \in U_\pi} \mathsf{MP}(\tilde{\mathcal{G}}^{\pi,\mu}, s_0, C)$.*

The algorithm's structure parallels the previous algorithm, with the main difference being that it solves an SG rather than an MDP.

*Safety.* For safety objectives, first the maximum probability to reach $B$ is computed via a QP with Player 2 as the maximizer. If said probability is $\geq 1 - p$, the problem reduces to solving an SSP problem in the transformed game $\tilde{\mathcal{G}}$, where the goal is to reach the target set $B$ with $\geq 1 - p$ probability. The SSP problem can be solved using VI or PI [3] [21]. The SSP problem usually assumes that the target sink (here it's $B$) is reached with probability 1 to ensure that the expected cost is finite. For our case, since there are zero cost paths always available, the same algorithm can be used. Here as soon as the play reaches $G$, the play can be stopped, and the cost is 0. The solution yields the transient breaking point and the frequency breaking point is 0. If the probability to reach $B$ is $< 1 - p$, it indicates that the strategy is not breakable.

[3]The current algorithms for SSP using VI and PI only give approximate solutions with convergence guarantees in the limit. We do not get the exact breaking point in this case, however, if exact solvers for SSP are found in the future, our approach can use them off-the-shelf.

*Reachability.* For reachability objectives, the algorithm first identifies the set of states in $\tilde{\mathcal{G}}$ from which Player 2 can ensure that the target set $G$ is avoided with probability 1. This is done via a backward fixed-point computation, starting with the set $E = S \setminus G$ and iteratively removing states according to the following rules until convergence:

- A Player 1 state is removed if there exists an action leading outside $E$ with positive probability.
- A Player 2 state is removed if all available actions lead outside $E$ with positive probability.

Next, we compute the maximum probability of reaching $E$ in $\tilde{\mathcal{G}}$ using QP. If the reachability probability is $< 1 - p$, the strategy cannot be broken, and the algorithm returns $(\natural, \natural)$. If the reachability probability is $\geq 1 - p$, we proceed to analyze the MECs contained in $E$. For each MEC, we compute its minimum mean payoff, which corresponds to the minimal average disturbance cost required to remain within that component indefinitely [8]. Let $B \subseteq E$ denote the set of MECs with zero mean payoff.

If $B$ is reachable with probability $\geq 1 - p$, the transient breaking point is finite and the frequency breaking point is 0, indicating that the strategy can be broken with finitely many disturbances. This is computed by solving an SSP problem in $\tilde{\mathcal{G}}$ with the target set $B$ and probability threshold $1 - p$. In the other case, the transient breaking point is $\omega$, and the frequency breaking point requires the use of weighted quotient stochastic game that is constructed by collapsing each MEC into a single state. A new terminal state $s_+$ is introduced and an outgoing transition from all collapsed states to $s_+$ is added, with the cost of this transition equal to the mean payoff of the corresponding MEC. The cost of all other transitions is set to zero. Finally, we solve an SSP problem in this quotient SG to compute the frequency breaking point.

**Theorem 2.** *Given an SGD $\mathcal{G}$, objective $\phi$, and pair of values $(t, f)$, deciding whether $\mathcal{EB}_{\mathcal{G},\phi} \geq (t, f)$ is as hard as solving SSP for SGs.*

# 5 WORST-CASE BREAKING POINT

In contrast to the expected case, we now consider the worst-case number of disturbances required to break a strategy over all possible paths. Now, the *worst-case transient breaking point* of a strategy is

$$\mathcal{B}^{T}_{\mathcal{G},\phi}(\pi) := \inf_{(\sigma,\delta) \in U_\pi} \inf \Big\{ x \in \langle \omega \rangle \;\big|\; P^{\pi,\sigma,\delta}_{\mathcal{G},s_0}\big(\rho \mid \mathcal{D}^{T}(\rho) \leq x\big) = 1 \Big\}.$$

We only require that almost all paths (i.e., with probability 1) have fewer than $k$ disturbances, reflecting the common probabilistic convention of ignoring measure-zero events that do not affect the typical behaviors. Recall again that when the set $U_\pi$ is empty, we define $\mathcal{B}^{T}_{\mathcal{G},\phi}(\pi)$ to be $\natural$, and when there exist no finite values in this set, we denote $\mathcal{B}^{T}_{\mathcal{G},\phi}(\pi)$ to be $\omega$. When the transient breaking point is $\omega$, we can compute the frequency of disturbances required. The *worst-case frequency breaking point* of a Player 1 strategy $\pi$ is

$$\mathcal{B}^{F}_{\mathcal{G},\phi}(\pi) := \inf_{(\sigma,\delta) \in U_\pi} \inf \Big\{ x \in [0,1] \;\big|\; P^{\pi,\sigma,\delta}_{\mathcal{G},s_0}\big(\rho \mid \mathcal{D}^{F}(\rho) \leq x\big) = 1 \Big\}.$$

We can now combine the two values and define the *worst-case breaking point* of a strategy.

Definition 4. *Given a SGD $\mathcal{G}$ and an objective $\phi$, the* worst-case breaking point *of a Player 1 strategy $\pi$ is given by the pair $\mathcal{B}_{\mathcal{G},\phi}(\pi) := \left(\mathcal{B}^T_{\mathcal{G},\phi}(\pi), \mathcal{B}^F_{\mathcal{G},\phi}(\pi)\right)$.*

## 5.1 Computing the Worst-Case Breaking Point

We describe the algorithm for transient and frequency worst-case breaking points for both safety and reachability objectives. We compute the breaking points for a memoryless strategy $\pi$ here, but it can easily be extended to finite-memory strategies by encoding the memory in the state space, similar to the expected case. First, we provide an overview of the key steps, deferring the detailed algorithms for computing the transient and frequency breaking points to later in this section. If $\phi$ is a safety objective, set $B$ is given as input. Otherwise, if $\phi$ is a reachability objective, we construct set $B$ as defined in Eq. (1). We then compute the maximum probability in the induced MDP $M_\pi$ of reaching $B$, using LP. We distinguish three cases: the probability is either $> 1-p$, $< 1-p$, or $= 1-p$.

**Case 1: Probability to reach $B$ is $> 1-p$.** In this case, the strategy is breakable with finitely many disturbances.

**Case 2: Probability to reach $B$ is $= 1-p$.** In this case, if the strategy is breakable in a finite number of disturbances, this number should be bounded by $k$: the number of disturbance actions present in the graph. We run the procedure to compute the worst-case transient breaking point for $|T^D|$ iterations. Based on the outcome of this procedure, it is further divided into two subcases:

a. If the procedure terminates at the $i^{th}$ iteration for some $i$, we have $\mathcal{B}^T_{\mathcal{G},\phi}(\pi) = i$.
b. If it does not terminate within $|T^D|$ iterations, it requires infinitely many disturbances, making the transient breaking point $\omega$ and the frequency breaking point 0. This is due to the fact that there are probabilistic loops that are required to reach $B$ with a $1-p$ probability, whereas the frequency of taking that action in the long term would still be 0.

**Case 3: Probability to reach $B$ is $< 1-p$.** Here, if $\phi$ is a safety objective, the strategy is not breakable with any amount of disturbances. If $\phi$ is a reachability objective, we compute the set $R$ using Eq. (2). If the probability of reaching $R \cup B$ is $< 1-p$, again, the strategy is not breakable with any amount of disturbances and the algorithm returns $(\natural, \natural)$. If the probability of reaching $R \cup B$ is $\geq 1-p$, the strategy is breakable but requires infinitely many disturbances. This results in the transient breaking point being $\omega$ and the frequency breaking point being computed using the procedure described later.

*5.1.1 Transient Breaking Point.* Given $k$,[4] we want to verify if $\mathcal{B}^T_{\mathcal{G},\phi}(\pi) \leq k$. We define a sequence of $k$ reachability LPs where the $i^{th}$ LP checks if the strategy $\pi$ can be broken using at most $i$ disturbances. The $i^{th}$ LP uses the solution of the $(i-1)^{st}$ LP and allows one more disturbance to compute the probability of reaching $B$ with at most $i$ disturbances. If the solution of $i^{th}$ LP is $\geq 1-p$, $\pi$ is breakable using $i$ disturbances. For completeness, the explicit LP can be found in the supplementary material.

[4]An upper bound here, if it is finite, can be computed by counting the number of iterations required by VI for MDPs to go beyond probability $1-p$ of reaching $B$.

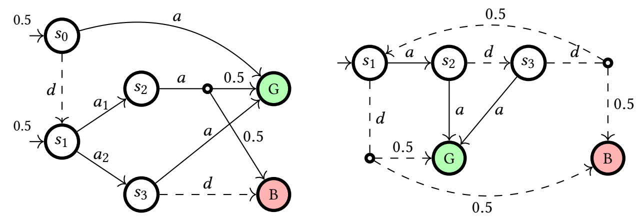


**Figure 5: An SGD where the most resilient $\pi$ must have memory even if $\delta$ is memoryless (left), and one where the optimal $\delta$ must rely on memory even if $\pi$ is memoryless (right).**

Each LP is solved in polynomial time, and each iteration only requires the result of the previous LP. This gives us an algorithm that is polynomial in terms of $|\mathcal{G}|$ and $k$. This gives us a parametrized polytime complexity where $k$ is given in unary, and in general a **PSPACE** algorithm. It can also be shown that the optimal disturbance strategy here may require memory of size $k$.

*5.1.2 Frequency Breaking Point.* Recall that the frequency breaking point is only computed when the specification is reachability and the probability of reaching $B$ is $< 1-p$ but the probability of reaching $R \cup B$ is $\geq 1-p$.

As in the expected case, the procedure assigns a frequency of disturbances required to stay within the set $B$ as 0 and assigns the mean payoff for the MECs in the set $R$ as their frequency. In contrast to the expected case, where we solved a SSP to find the final expected value, we need to consider the worst case. Here, we iteratively remove a MEC with the highest disturbance frequency. After each removal, it recomputes the probability of reaching the remaining MECs in $R \cup B$. This process continues until the probability of reaching the remaining MECs drops below $1-p$. The frequency of disturbances required to remain in the last removed MEC is then returned as the frequency breaking point.

**Theorem 3.** *Given an SGD $\mathcal{G}$, objective $\phi$, Player 1 strategy $\pi$, and pair of values $(t, f)$, deciding if $\mathcal{B}_{\mathcal{G},\phi}(\pi) \geq (t, f)$ is in* **PSPACE**.

## 5.2 Optimal Strategy: Worst-Case Breaking Point

An optimally resilient strategy is a strategy that achieves the following worst-case breaking point:

$$\mathcal{B}_{\mathcal{G},\phi} = \left(\max_{\pi} \mathcal{B}^T_{\mathcal{G},\phi}(\pi), \max_{\pi} \mathcal{B}^F_{\mathcal{G},\phi}(\pi)\right)$$

We first discuss memory requirements for transient and frequency breaking points, and later provide the algorithm to compute them.

*5.2.1 Memory Requirements for Transient Breaking Point.* Making use of the reduction to standard stochastic games from Section 3.3, we next show that deciding the worst-case transient breaking point is equivalent to solving the unfolded stochastic game. This allows us to use results concerning stationarity in standard SGs to reason about memory requirements in SGDs.

**Lemma 5.** *Let $\mathcal{G}$ be an SGD and $\mathcal{G}^{\dagger k}$ its corresponding $k$-unfolded stochastic game, for some $k \in \langle\omega\rangle$. Then, for all Player 1 strategies $\pi$, we have $\mathcal{B}^T_{\mathcal{G},\phi}(\pi) \leq k$ if and only if $\sup_{\tilde{\sigma}} \mathbb{P}^{\tilde{\pi},\tilde{\sigma}}_{\mathcal{G}^{\dagger k},(s_0,k)}(\neg\phi) \geq 1-p$.*

If we let $k$ be the optimal breaking point, $k = \max_{\pi \in \Pi} \mathcal{B}^T_{\mathcal{G},\phi}(\pi)$, then the unfolded game $\mathcal{G}^{\dagger k}$ is a standard stochastic game, and the optimal strategies for both players are therefore memoryless [26]. Lemma 5, together with Lemma 2, implies that an SGD $\mathcal{G}$ is stationary with respect to the state-counter pair $(s, i)$. Here, $i$ represents the number of disturbances remaining, and the pair $(s, i)$ uniquely identifies a state in the corresponding unfolded game. Consequently, we can derive the following two corollaries.

**Corollary 1.** *For any SGD $\mathcal{G}$, there exists an optimally resilient Player 1 strategy of the form $\pi^* : S \times \langle k+1 \rangle \to A$.*

**Corollary 2.** *For any SGD $\mathcal{G}$, there exists an optimal $k$-disturbance strategy of the form $\delta^* : S \times \langle k+1 \rangle \to A$.*

In the next two examples, we construct SGDs where the optimal strategies require memory.

**Example 1.** *Let $\mathcal{G}$ be as in Figure 5 (left) with $\phi = P_{\geq 0.75}(\Diamond G)$. There is a step-counting strategy of the form $\pi^* : S \times \langle 2 \rangle \to A$ with a worst-case transient breaking point of 2: $\pi^*(s_0, \_) = a$, $\pi^*(s_1, 0) = d$, $\pi^*(s_1, 1) = a$, $\pi^*(s_3, \_) = a$, and $\pi^*(s_2, \_) = a$. In contrast, no memoryless Player 1 strategy can achieve a breaking point of 2.*

**Example 2.** *Let $\mathcal{G}$ be as in Figure 5 (right) with $\phi = P_{\geq 0.5}(\Diamond G)$. Define the memoryless Player 1 strategy $\pi(s) = a$ for every state $s$. Then, $\pi$ has the worst-case transient breaking point of 3, as the following step-counting 3-disturbance strategy breaks it: $\delta(s_1, 3) = \bot$, $\delta(s_1, 2) = \delta(s_1, 1) = d$, $\delta(s_2, \_) = d$, $\delta(\_, 0) = \bot$. Yet, no memoryless disturbance strategy that disturbs at most 3 times can break $\pi$.*

In general, it is *necessary* for optimal strategies to perform step counting (i.e., their memory is of the form $S \times \langle k+1 \rangle$), as seen in Examples 1 and 2, while Corollaries 1 and 2 show this is *sufficient*. Thus, we always assume that optimal strategies have this form in the case of worst-case transient resilience without loss of generality.

*5.2.2 Memory Requirements for Frequency Breaking Point.* In contrast to the transient case, memoryless strategies are sufficient for the frequency breaking point. We discuss this for reachability objectives; in the case of safety objectives either it can be broken with finite disturbances or it cannot be broken at all.

**Lemma 6.** *If $\phi$ is reachability, and it is not possible to break the strategy with finite disturbances, then the optimal disturbance strategy is memoryless.*

The intuition behind Lemma 6 is that, since the strategy cannot be broken with finite disturbances, the disturbance strategy must rely on staying in some MECs of the MDP induced by the Player 1 strategy. In these MECs, the optimal disturbance strategy to minimize the mean payoff is memoryless. Thus, the overall disturbance strategy can be chosen to be memoryless.

*5.2.3 Computing Optimally Resilient Strategies.* The procedure here is similar to the procedure in Section 4.2. Here also, we use the transformation to $\tilde{\mathcal{G}}$. If $\phi$ is safety, then the probability of reaching $B$ is computed. If it is $< 1 - p$, then the strategy is not breakable. In the other case, the frequency breaking point is 0 and the transient breaking point is computed as described later in this section.

If $\phi$ is reachability, then the algorithm computes the set of states $E$ from which Player 2 can force the play to never reach $G$. This can be computed using the fix-point computation as described in Section 4.2. If the maximum probability of reaching $E$ is $< 1 - p$, then the strategy is again not breakable. If it is $\geq 1 - p$, the minimum mean payoff of each MEC in $E$ is computed, which gives us the minimum frequency of disturbances required to stay in that MEC.

Let $B$ be the set of MECs in $E$ with 0 mean payoff and if $B$ is reachable with probability $\geq 1 - p$, then the frequency breaking point is 0 and the transient breaking point is computed as described next. Otherwise, the algorithm iteratively removes a MEC with the highest disturbance frequency. After each removal, it recomputes the probability of reaching the remaining MECs in $E$. This process continues until the probability of reaching the remaining MECs drops below $1 - p$. The frequency required to remain in the last removed MEC is then returned as the frequency breaking point.

*5.2.4 Transient Breaking Point.* Given $k$, we want to verify whether $\mathcal{B}^T_{\mathcal{G},\phi} \leq k$. We define a sequence of $k$ QPs where the $i^{th}$ QP checks if all strategies $\pi$ can be broken using at most $i$ disturbances. Here too, the $i^{th}$ QP uses the solution of the $(i-1)^{th}$ QP, and by allowing one more disturbance, it computes the probability of reaching $B$ with $i$ disturbances. Intuitively, the $i^{th}$ QP solves for the maximum probability of reaching bad states using the method in [10] in the game unfolded for $i$ disturbances (Section 3.3) but reuses results from the game unfolded for $i - 1$ disturbances. Description of these QPs is provided in the Supplementary Material.

Note that the decision problem corresponding to these specific QPs can be solved in **NP** [10], but in our case we also need to find the solution of the QP. This can be extracted using polynomially many **NP** queries, where in $j^{th}$ query we will ask whether the $j^{th}$ bit of the solution is 1. Also, in each iteration, we only need to remember the solution of the last iteration. Thus, we can solve this in polynomial time with an **NP** oracle if $k$ is given in unary. In practice, the **NP**-oracle complexity would involve querying a SAT solver polynomially many times to extract the solution bit-by-bit, and we would expect the state-of-the-art SAT-solvers to scale well.

**Theorem 4.** *Given an SGD $\mathcal{G}$, an objective $\phi$, and a pair of values $(t, f)$, deciding whether $\mathcal{B}_{\mathcal{G},\phi} \geq (t, f)$ is in* **PSPACE**.

## 6 CONCLUSIONS AND FUTURE WORK

In this work, we explored the concept of resilient strategies in stochastic systems to analyze their robustness against disturbances. We introduced novel formulations for resilience by considering both expected and worst-case breaking points, and refined the notion of resilience through measures such as frequency-based disturbances. We provided algorithms for computing resilience, offering solutions for both expected and worst-case scenarios, and highlighted the trade-offs between optimality and resilience. By introducing this mathematical framework, as well as providing proofs of properties of general theoretical interest, our work can serve as a foundation for developing practical solutions to real-world problems. Future work includes extending these concepts to partially observable Markov decision processes (POMDPs), multi-agent settings, and alternative objectives beyond reachability and safety.

## ACKNOWLEDGMENTS

Kush Grover was partially supported by the STEP-RL project funded by the European Research Council under GA number 101115870. Nils Jansen was partially supported by the European Research Council Starting Grant 101077178 (DEUCE). Debraj Chakraborty was partially supported by the National Research Foundation, Singapore, under its RSS Scheme (NRF-RSS2022-009). Jan Křetínský was partially funded by German Research Foundation (DFG) - 427755713 for the project titled "Group-By Objectives in Probabilistic Verification (GOPro)". Jan Křetínský, Kush Grover, and Debraj Chakraborty were partially supported by the MUNI Award in Science and Humanities MUNI/I/1757/2021 of the Grant Agency of Masaryk University, and the Intelligence-Oriented Verification & Controller Synthesis (InOVationCS) project funded by the European Research Council under GA number 101171844.


## REFERENCES


[1] Gul Agha and Karl Palmskog. 2018. A survey of statistical model checking. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 28, 1 (2018), 1–39.

[2] Pranav Ashok, Krishnendu Chatterjee, Przemysław Daca, Jan Křetínský, and Tobias Meggendorfer. 2017. Value Iteration for Long-Run Average Reward in Markov Decision Processes. In *Computer Aided Verification*, Rupak Majumdar and Viktor Kunčak (Eds.). Springer International Publishing, Cham, 201–221.

[3] Pranav Ashok, Mathias Jackermeier, Pushpak Jagtap, Jan Křetínskỳ, Maximilian Weininger, and Majid Zamani. 2020. dtControl: decision tree learning algorithms for controller representation. In *Proceedings of the 23rd international conference on hybrid systems: Computation and control*. 1–7.

[4] Pranav Ashok, Jan Křetínskỳ, and Maximilian Weininger. 2019. PAC statistical model checking for Markov decision processes and stochastic games. In *International Conference on Computer Aided Verification*. Springer, 497–519.

[5] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of model checking.* MIT Press.

[6] Alexander Benvenuti, Calvin Hawkins, Brandon Fallin, Bo Chen, Brendan Bialy, Miriam Dennis, and Matthew Hale. 2024. Differentially Private Reward Functions for Markov Decision Processes. In *2024 IEEE Conference on Control Technology and Applications (CCTA)*. 631–636. https://doi.org/10.1109/CCTA60707.2024.10666610

[7] Roderick Paul Bloem, Rüdiger Ehlers, Swen Jacobs, and Robert Könighofer. 2014. How to Handle Assumptions in Synthesis. In *Proceedings 3rd Workshop on Synthesis (SYNT 2014) (EPTCS)*. Electronic Proceedings in Theoretical Computer Science, 34–50. https://doi.org/10.4204/EPTCS.157.7 Workshop on Synthesis (SYNT 2014) ; Conference date: 23-07-2014 Through 24-07-2014.

[8] Endre Boros, Khaled Elbassioni, Vladimir Gurvich, and Kazuhisa Makino. 2017. A convex programming-based algorithm for mean payoff stochastic games with perfect information. *Optimization Letters* 11, 8 (2017), 1499–1512. https://doi.org/10.1007/s11590-017-1140-y

[9] Anne Condon. 1990. On Algorithms for Simple Stochastic Games. *Advances in computational complexity theory* 13 (1990), 51–72.

[10] Anne Condon. 1992. The complexity of stochastic games. *Information and Computation* 96, 2 (1992), 203–224.

[11] Eric Dallal, Daniel Neider, and Paulo Tabuada. 2016. Synthesis of safety controllers robust to unmodeled intermittent disturbances. In *2016 IEEE 55th Conference on Decision and Control (CDC)*. 7425–7430. https://doi.org/10.1109/CDC.2016.7799416

[12] Parham Gohari, Matthew Hale, and Ufuk Topcu. 2020. Privacy-Preserving Policy Synthesis in Markov Decision Processes. In *2020 59th IEEE Conference on Decision and Control (CDC)* (Jeju Island, Korea (South)). IEEE Press, 6266–6271. https://doi.org/10.1109/CDC42340.2020.9304015

[13] Kush Grover, Markel Zubia, Debraj Chakraborty, Muqsit Azeem, Nils Jansen, and Jan Kretinsky. 2026. Resilient Strategies for Stochastic Systems: How Much Does It Take to Break a Winning Strategy? arXiv:2602.24191 [cs.GT] https://arxiv.org/abs/2602.24191

[14] C Hopps. 2000. Rfc2992: Analysis of an equal-cost multi-path algorithm.

[15] Jan Křetínský, Tobias Meggendorfer, Maximilian Prokop, and Sabine Rieder. 2023. Guessing winning policies in LTL synthesis by semantic learning. In *International Conference on Computer Aided Verification*. Springer, 390–414.

[16] Isabella Kuo and Valdinei Freire. 2021. Probability-to-Goal and Expected Cost Trade-Off in Stochastic Shortest Path. In *Computational Science and Its Applications – ICCSA 2021*, Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Chiara Garau, Ivan Blečić, David Taniar, Bernady O. Apduhan, Ana Maria A. C. Rocha, Eufemia Tarantino, and Carmelo Maria Torre (Eds.). Springer International Publishing, Cham, 111–125.

[17] Zhengshang Liu, Yue Yang, Tim Miller, and Peta Masters. 2021. Deceptive Reinforcement Learning for Privacy-Preserving Planning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems* (Virtual Event, United Kingdom) *(AAMAS '21)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 818–826.

[18] Alberto Marchesi and Nicola Gatti. 2021. Trembling-Hand Perfection and Correlation in Sequential Games. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 6 (May 2021), 5566–5574. https://doi.org/10.1609/aaai.v35i6.16700

[19] Daniel Neider, Patrick Totzke, and Martin Zimmermann. 2020. Optimally Resilient Strategies in Pushdown Safety Games. In *45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, August 24-28, 2020, Prague, Czech Republic (LIPIcs, Vol. 170)*, Javier Esparza and Daniel Král' (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 74:1–74:15. https://doi.org/10.4230/LIPICS.MFCS.2020.74

[20] Daniel Neider, Alexander Weinert, and Martin Zimmermann. 2020. Synthesizing optimally resilient controllers. *Acta Informatica* 57, 1-2 (2020), 195–221. https://doi.org/10.1007/S00236-019-00345-7

[21] Stephen D Patek and Dimitri P Bertsekas. 1999. Stochastic shortest path games. *SIAM Journal on Control and Optimization* 37, 3 (1999), 804–824.

[22] Shashank Pathak, Erika Ábrahám, Nils Jansen, Armando Tacchella, and Joost-Pieter Katoen. 2015. A Greedy Approach for the Efficient Repair of Stochastic Models. In *NFM (Lecture Notes in Computer Science, Vol. 9058)*. Springer, 295–309.

[23] Vern Paxson. 1997. End-to-end Internet packet dynamics. In *Proceedings of the ACM SIGCOMM'97 conference on Applications, technologies, architectures, and protocols for computer communication*. 139–152.

[24] Martin L. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (1st ed.). John Wiley & Sons, Inc., USA.

[25] Stanly Samuel, Kaushik Mallik, Anne-Kathrin Schmuck, and Daniel Neider. 2020. Resilient Abstraction-Based Controller Design. In *2020 59th IEEE Conference on Decision and Control (CDC)*. 2123–2129. https://doi.org/10.1109/CDC42340.2020.9303932

[26] Lloyd S Shapley. 1953. Stochastic games. *Proceedings of the national academy of sciences* 39, 10 (1953), 1095–1100.

[27] Moshe Y. Vardi. 2020. Efficiency vs. resilience: what COVID-19 teaches computing. *Commun. ACM* 63, 5 (April 2020), 9. https://doi.org/10.1145/3388890

[28] Pian Yu, Shufang Zhu, Giuseppe De Giacomo, Marta Kwiatkowska, and Moshe Vardi. 2024. The Trembling-Hand Problem for LTLf Planning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Kate Larson (Ed.). International Joint Conferences on Artificial Intelligence Organization, 3631–3641. https://doi.org/10.24963/ijcai.2024/402 Main Track.

[29] Yuanhui Zhang, Haipeng Wu, Brian T Denton, James R Wilson, and Jennifer M Lobo. 2019. Probabilistic sensitivity analysis on Markov models with uncertain transition probabilities: an application in evaluating treatment decisions for type 2 diabetes. *Health Care Manag. Sci.* 22, 1 (March 2019), 34–52.