Beyond strong-cyclic: doing your best in stochastic environments

Benjamin Aminof[1] , **Giuseppe De Giacomo**[2] , **Sasha Rubin**[3] and **Florian Zuleger**[1]
[1]TU Vienna, Austria
[2]University of Rome "La Sapienza", Italy
[3]University of Sydney, Australia
{aminof, zuleger}@forsyte.at, degiacomo@diag.uniroma1.it, sasha.rubin@sydney.edu.au

Abstract

"Strong-cyclic policies" were introduced to formalize trial-and-error strategies and are known to work in Markovian stochastic domains, i.e., they guarantee that the goal is reached with probability 1. We introduce "best-effort" policies for (not necessarily Markovian) stochastic domains. These generalize strong-cyclic policies by taking advantage of stochasticity even if the goal cannot be reached with probability 1. We compare such policies with optimal policies, i.e., policies that maximize the probability that the goal is achieved, and show that optimal policies are best-effort, but that the converse is false in general. With this framework at hand, we revisit the foundational problem of what it means to plan in nondeterministic domains when the nondeterminism has a stochastic nature. We show that one can view a nondeterministic planning domain as a representation of infinitely many stochastic domains with the same support but different probabilities, and that for temporally extended goals expressed in LTL/LTLf a finite-state best-effort policy in one of these domains is best-effort in each of the domains. In particular, this gives an approach for finding such policies that reduces to solving finite-state MDPs with LTL/LTLf goals. All this shows that "best-effort" policies are robust to changes in the probabilities, as long as the support is unchanged.

1 Introduction

Planning in nondeterministic environments is one of the key challenges of AI: often the agent does not have control over everything that happens in its environment so it views the environment as nondeterministic (in a devilish sense), i.e., only partially controllable by the agent itself. This topic has developed as a subarea of Planning on Fully Observable Nondeterministic Domains (FOND) [Rintanen, 2004; Geffner and Bonet, 2013; Ghallab *et al.*, 2016; Haslum *et al.*, 2019] and numerous solvers are available [Mattmüller *et al.*, 2010; Kissmann and Edelkamp, 2011; Muise *et al.*, 2012; Ramírez and Sardiña, 2014; Geffner and Geffner, 2018].

The direct extension of classical planning to FOND gives rise to *strong plans*, i.e., policies or strategies that tell the agent what to do in every situation and that achieve the goal in spite of the nondeterministic environment's response to the agent's actions [Cimatti *et al.*, 1998; Pistore and Traverso, 2001]. However, in many situations these kinds of plans do not exist. So, a weaker notion of plan was introduced, the *strong cyclic plans* [Daniele *et al.*, 1999; Cimatti *et al.*, 2003]. Intuitively, these policies allow loops from which the agent assumes it will eventually exit, e.g., they may encode trial-and-error strategies. Interestingly, as recently noted in [D'Ippolito *et al.*, 2018], the fundamental question of formalizing "contexts under which these type[s] of plans will indeed achieve the objectives... has not received much attention and has mostly been discussed informally". A first attempt to characterize these contexts has been to make a *logical fairness* assumption on the actions effects, see [D'Ippolito *et al.*, 2018] for a discussion. Unfortunately, while this solves the problem for classical reachability goals, if we consider temporally extended goals, e.g., expressed in LTL or LTL_f, this characterization fails because the simple fairness assumption that is sufficient for reachability is not adequate for LTL or LTL_f goals [Aminof *et al.*, 2020b]. An alternative to logical fairness is *stochastic fairness* (see [Aminof *et al.*, 2020b] for a detailed comparison) already suggested in early papers, e.g., [Cimatti *et al.*, 2003]. This point of view assumes that the nondeterminism in FOND arises from a stochastic nature of the environment: the environment is not adversarial to the agent, but chooses its effects according to some unknown probability distribution. In particular, strong-cyclic policies rely on the fact that in a stochastic environment, repeating an action (in the same situation) will eventually lead to the desired effect.

The question now arises: on what objects is this unknown probability distribution given? Is it given on the current values of the fluents and the action, and if so, what is the justification for that? Why should we assume such a probability distribution is stationary with respect the domain states? After all, the nondeterministic domain describes only the *support*, i.e., the effects with non-zero probability. No further assumptions should be made on the effect distributions. In this paper we take this observation seriously.

We focus on stochastic domains and LTL and LTL_f goals. We study three classes of policies: (*i*) those that *almost-*

surely enforce the goal, i.e., with probability 1, [Baier *et al.*, 2008], (*ii*) those that are *optimal* wrt enforcing the goal [Puterman, 2005], and (*iii*) those that are *best-effort* wrt enforcing the goal [Aminof *et al.*, 2021b]. The latter, which we call *stochastic best-effort policies*, are novel in the context of stochastic domains. Such a policy will guarantee that even if the goal is not achievable with probability 1, it will, intuitively, seize every opportunity to achieve the goal. For example, suppose an agent is offered some money that, if it were to accept, can only be used to place one of two bets, Bet 1 or Bet 2, each of which gives it a non-trivial probability of winning; however, the agent does not know these probabilities, or even which probability is larger. If the agent's goal is to win money at the casino, what actions should it take? In this setting the agent cannot guarantee with probability 1 that it will win a bet. Also, since the relative probabilities of winning each bet are not known, it can't deduce an optimal policy either. A rational solution would be to accept the money, rather than refuse the money, and place either bet. This is achieved by a stochastic best-effort policy.

In Section 3, we show the following: optimal policies are stochastic best-effort, but a stochastic best-effort policy need not be optimal; if an almost-sure policy exists, the stochastic best-effort policies are exactly the almost-sure policies; stochastic best-effort policies always exist, while almost-sure or optimal policies may not exist.

In Section 4, we consider a FOND as a specification of a set of stochastic domains, i.e., those whose support is specified by the FOND. We show that a finite-state policy that is stochastic best-effort in one of these stochastic domains is in fact stochastic best-effort in all of these stochastic domains. This immediately provides an algorithm to compute a single stochastic best-effort policy that works for each of the stochastic domains induced by the FOND: choose a stochastic domain compliant with the FOND in the most convenient way (e.g., assigning uniform probability to all nondeterministic transition of the FOND), and compute a (finite-state) optimal policy for that particular stochastic domain. Note that the stochastic domain can be chosen to be an MDP, which always has an optimal policy. This policy is in fact stochastic best-effort for every stochastic domain specified by the FOND. Moreover, if the policy almost-surely enforces the goal in that stochastic domain then it almost-surely enforces the goal in every stochastic domain specified by the FOND.

In other words, we show that the true nature of FOND for strong cyclic-plans (or FOND under "stochastic fairness") is that of a *generalized planning problem* [Srivastava, 2010; Bonet *et al.*, 2010; Hu and De Giacomo, 2011; Bonet *et al.*, 2017] over the set of stochastic domains induced by the FOND. Moreover, our studies show that even if the strong cyclic solution does not exist, instead of giving up, we can provide a best-effort solution anyway, which unlike optimal solutions is robust in that it is shared by all induced stochastic domains. Observe that this last aspect is of interest per se since it can also be understood as follows: even if the probabilities in a stochastic domain, such as an MDP, are not accurate for some reason, as long as the support is correct, the nominal optimal solution, although not optimal in reality remains best-effort anyway.

2 Preliminaries

Notation

Sequences may be written $(x_0, x_1, \cdots)$ or $x_0 x_1 \cdots$. If h is a finite sequence then $last(h)$ is its last element. If h is a prefix of h' we say that h' *extends* h. A set is *countable* if it is finite or countably infinite. Let $Dbn(X)$ denote the set of *distributions* over X, i.e., functions $d : X \to [0, 1]$ such that $\sum_{x \in X} d(x) = 1$. An element x is in the *support* of d if $d(x) > 0$.

Stochastic Domains

Let F be a finite set of Boolean variables called *fluents*. We write $Obs = 2^F$ for the set of evaluations of the fluents, sometimes called *observations* or *percepts*. For symmetry, let A be a finite set of Boolean variables (disjoint from F) and let $Act = 2^A$ be the set of *actions*. A *stochastic domain* D is a tuple (F, A, s_0, Pr) where

1. $s_0 \in Obs$ is the *initial* observation, and
2. $Pr : Hist \times Act \to Dbn(Obs)$ is the *transition function*. Here $Hist$ is the set of *histories*, i.e., sequences $(s_0, a_0, s_1, a_1, \cdots, s_k)$ that start with the initial observation, alternate observations and actions, and end in an observation.

The *support function* of D is the function $\Delta : Hist \times Act \to 2^{Obs}$ defined by $s \in \Delta(h, a)$ iff $Pr(h, a)(s) > 0$. If $Pr(h, a)(s) > 0$ we say that s is in the *support* of $Pr(h, a)$. We say that D has *Markovian support* if $\Delta(h, a)$ only depends on $last(h)$ and a, i.e., $last(h) = last(h')$ implies that $\Delta(h, a) = \Delta(h', a)$. In this case, we may write the support function as $\Delta : Obs \times Act \to 2^{Obs}$. We say that D is *bounded* if there is some $\epsilon > 0$ such that $Pr(h, a)(s) \geq \epsilon$ for all h, a and s in the support of $Pr(h, a)$. A domain is *Markovian* if $Pr(h, a)$ only depends on $last(h)$ and a, i.e., $last(h) = last(h')$ implies that $Pr(h, a) = Pr(h', a)$.

Note that finite-state Markov Decision Processes (MDPs) without rewards are exactly the Markovian stochastic domains. Some of the counterexample stochastic-domains that we give are MDPs; in the figures, we will label their transitions by $a : r$ to mean that, given that action a was played, the probability of that transition is r.

Plays and Policies

Let $D = (F, A, s_0, Pr)$ be a stochastic domain. A *play* of D is an infinite sequence $\pi = (s_0, a_0, s_1, a_1, \cdots)$ starting in the initial observation. Let $Plays$ denote the set of all plays in D. A *policy* (aka strategy, plan) is a function $\sigma : Hist \to Act$. A σ-play (resp. σ-history) is such that $\sigma(s_0, a_0, s_1, a_1, \cdots, s_t) = a_t$ for every t. A policy is *finite-state* if it can be represented as a finite-state input/output automaton that, on reading $h \in Hist$ as input, outputs the action $\sigma(h)$. A policy is *memoryless* if $last(h) = last(h')$ implies that $\sigma(h) = \sigma(h')$. In this case we can write $\sigma : Obs \to Act$.

Markov chains

A *Markov chain* M is a tuple (Q, s_0, p) where Q is a countable set of *states*, $s_0 \in Q$ is the *initial state*, and $p : Q \to Dbn(Q)$. We write $p(s, t)$ for $p(s)(t)$, and call these the *transition probabilities* of M. The graph (Q, E_M) *induced* by M

has state set Q and edge set E_M consisting of pairs (q, q') such that q' is in the support of $p(q)$. Thus, graph-theoretic notions (like connected components) are defined for M by considering the induced graph. A Markov chain is *finite* if Q is finite. A Markov chain M induces a canonical probability space on the set of infinite paths in M starting in s_0 that we denote (Ω_M, Alg_M, μ_M), e.g., [Puterman, 2005]. We may drop subscripts for readability. Elements of Alg are called *events*. All sets that we measure are either assumed to be events, or can be shown to be events, although we may not explicitly say so. For a history h we denote by C_h the event consisting of the plays with prefix h. If $\mu(E) = 1$ we say that *almost surely E holds (in M)*. In case $\mu(F) > 0$ define $\mu(E|F) := \mu(E \cap F)/\mu(F)$.

Probability Measures

Fix a domain D and a policy σ. Define the Markov chain $D[\sigma] = (Hist, s_0, p)$ where $p(h, h') := Pr(h, \sigma(h))(s)$ for $h' = h\sigma(h)s$. The measure of the induced probability space is denoted $\mu_{D[\sigma]}$. Note that a play in $D[\sigma]$ is a sequence of histories of D, i.e., of the form $(s_0), (s_0, a_0, s_1), (s_0, a_0, s_1, a_1, s_2), \cdots$, whose limit is a play in D, i.e., of the form $(s_0, a_0, s_1, a_1, \cdots)$. Thus, for convenience, we will refer to σ-plays in D as plays in $D[\sigma]$, and vice versa. Intuitively, $\mu_{D[\sigma]}(E)$ is the probability that a play in D starting from s_0 generated by σ is in E. We say that σ *almost surely enforces E (in D)* if $\mu_{D[\sigma]}(E) = 1$.

Linear-time Temporal Logic (LTL/LTLf)

The *formulas of* LTL a finite set AP of atoms are defined by the following BNF (where $p \in AP$): $\varphi ::= p \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathsf{X}\varphi \mid \varphi \,\mathsf{U}\, \varphi$. We use the usual abbreviations, $\mathsf{F}\,\varphi \doteq \mathsf{true}\,\mathsf{U}\,\varphi$, $\mathsf{G}\,\varphi \doteq \neg\,\mathsf{F}\neg\varphi$. A *trace* τ is an infinite sequence of valuations of the atoms, i.e., $\tau \in (2^{AP})^\omega$. For $n \geq 0$, write τ_n for the valuation at position n. Given τ, n, and φ, the satisfaction relation $(\tau, n) \models \varphi$, stating that φ holds at step n of the sequence τ, is defined as follows: 1. $(\tau, n) \models p$ iff $p \in \tau_n$; 2. $(\tau, n) \models \varphi_1 \vee \varphi_2$ iff $(\tau, n) \models \varphi_1$ or $(\tau, n) \models \varphi_2$; 3. $(\tau, n) \models \neg\varphi$ iff $(\tau, n) \models \varphi$ does not hold; 4. $(\tau, n) \models \mathsf{X}\,\varphi$ iff $(\tau, n + 1) \models \varphi$; and 5. $(\tau, n) \models \varphi_1 \,\mathsf{U}\, \varphi_2$ iff there exists $m \geq n$ such that: $(\tau, m) \models \varphi_2$ and $(\tau, j) \models \varphi_1$ for all $n \leq j < m$. If $\tau, 0 \models \psi$ we write $\tau \models \psi$ and say that τ *satisfies* ψ and that τ is a *model* of ψ. In case $AP = F \cup A$, we say that a *play* $(s_0, a_0, s_1, a_1, \cdots)$ *satisfies* φ if the corresponding trace $(s_0 \cup a_0, s_1 \cup a_1, \cdots)$ satisfies φ. We also consider the variant LTL_f (Bacchus and Kabanza [2000], Baier and McIlraith [2006], De Giacomo and Vardi [2013]). It has the same syntax and semantics as LTL except that τ is a finite sequence and X and U are redefined as follows: for $n < |\tau|$ define $\tau, n \models \mathsf{X}\,\psi$ if $n < |\tau| - 1$ and $\tau, n + 1 \models \psi$; define $\tau, n \models \varphi_1 \,\mathsf{U}\, \varphi_2$ iff there exists m with $n \leq m < |\tau|$ such that $(\tau, m) \models \varphi_2$ and $(\tau, j) \models \varphi_1$ for all $n \leq j < m$.[1]

We use the following convention for interpreting LTL_f formulas over infinite traces: if τ is infinite and ψ is an LTL_f formula, then $\tau \models \psi$ means that some finite prefix of τ satisfies ψ (analogous to the agent using an explicit "stop" action).

[1] "Optimistic" variants of these operators can be defined, e.g., "weak next ψ" holds if $n < |\tau| - 1$ implies $\tau, n + 1 \models \psi$.

Goals and Planning Problems

A *planning problem* (D, G) consists of a stochastic domain D and a set G of plays, called a *goal*. We say that a play *satisfies* G if it is in G. We only require that G satisfies the following technical property: for every policy σ, the set of plays of $D[\sigma]$ that are in G is measurable. This includes goals represented by LTL/LTL_f formulas and by automata $\mathcal{M}$. If D is stochastic, we will write $\mu_{D[\sigma]}(G)$ for the measure of the event consisting of all plays in $D[\sigma]$ that are in G.

3 Stochastic best-effort (SBE) policies

In this section we provide the definition of stochastic best-effort policy, and compare it to optimal policies. Fix a stochastic domain D and a goal G. For a policy σ, and σ-history h, define $val_{D,G}(\sigma, h)$ (simply $val(\sigma, h)$) as follows:

$$val_{D,G}(\sigma, h) := \begin{cases} +1 & \text{if } \mu_{D[\sigma]}(G|C_h) = 1 \\ -1 & \text{if } \mu_{D[\sigma]}(G|C_h) = 0 \\ 0 & \text{otherwise.} \end{cases}$$

The following fact about the extreme values follows from additivity of $\mu_{D[\sigma]}$:

Proposition 1 (Hereditary)**.** *If $val(\sigma, h) \neq 0$, and h' is σ-history that extends h, then $val(\sigma, h) = val(\sigma, h')$*

Here is the main definition of this work:

Definition 1 (Stochastic best-effort – SBE)**.** *Fix a stochastic domain D and a goal G. A policy σ is called* stochastic best-effort (SBE) *for the planning problem (D, G) if for every σ-history h of D:*

$$val_{D,G}(\sigma, h) = \max_{\sigma'} \{val_{D,G}(\sigma', h) : h \text{ is a } \sigma'\text{-history}\}.$$

Intuitively, an SBE policy tries, from every history, to almost-surely enforce G, and if this is not possible, then at least tries to ensure that G holds with positive probability. Thus, even if there is no policy that almost-surely enforces G from the start, the coin tosses from the environment may lead to such an opportunity in the future, and an SBE policy will take advantage of this. By maintaining a positive probability where possible, SBE policies do not close the door on opportunities that may arise due to stochasticity. Moreover, differentiating between probability zero, non-zero, and one, turns out to be robust even if very little is known of the probabilities themselves, but only what is and is not possible, see Section 4. It turns out that SBE policies always exist!

Theorem 1. *There exists a SBE policy for (D, G).*

The proof of this result is an adaptation of the fact that best-effort policies exist in the context of synthesis under assumptions [Aminof *et al.*, 2020a]. Moreover, the special case of temporal goals and bounded stochastic domains with Markovian support will follow from Theorem 4 Part 1.

3.1 Relation to optimization problems

In the case that probabilities in a stochastic domain are known or can be sampled, the natural problem is to find optimal policies, i.e., ones that maximize the probability of success. A policy σ is *optimal* for the planning problem (D, G) if $\sigma = \text{argsup}_{\sigma'} \mu_{D[\sigma']}(G)$. It is interesting to note that, in contrast to SBE policies, optimal policies don't always exist, even for reachability goals:

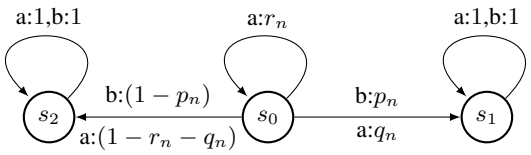

Figure 1: Generic planning problem. The starting observation is s_0, there are two actions "a" and "b", and the goal G is "eventually s_1". The parameter $n = 0, 1, 2, \cdots$ is the number of time steps that have passed. The variables r_n, p_n, q_n are instantiated in the text.

Example 1 (Optimal need not exist)**.** Consider Fig. 1 where $p_n := (n+1)/(2n+3)$ and $r_n := 1$ (and thus $q_n := 0$). One can calculate that for every policy σ, $\mu_{D[\sigma]}(G) < 1/2$, while $\sup_\sigma \mu_{D[\sigma]}(G) = 1/2$. We remark that D is bounded and has Markovian support.

Optimal policies are stochastic best-effort:

Theorem 2 (Optimal $\rightarrow$ SBE)**.** *Let D be a stochastic domain, and let G be a goal. Then every optimal policy σ for (D, G) is stochastic-best effort for (D, G).*

Proof. Suppose σ is not stochastic best-effort. Then $val(\sigma, h) < val(\sigma', h)$ for some σ' and some history h that is both a σ-history and a σ'-history. There are two cases depending on $val(\sigma, h) \in \{-1, 0\}$. We consider the case that $val(\sigma, h) = 0$. In this case, $\mu_{D[\sigma]}(G|C_h) = 0$ and $\mu_{D[\sigma']}(G|C_h) > 0$. Define the policy $\sigma'' := \sigma[h \leftarrow \sigma']$, which agrees with σ' on h and all of its extensions, and otherwise agrees with σ. Then $\mu_{D[\sigma'']}(G|C_h) > 0$, and thus $\mu_{D[\sigma'']}(G) > \mu_{D[\sigma]}(G)$. Thus σ is not optimal. The case that $val(\sigma, h) = -1$ is similar. □

We point out that the converse is false already for reachability goals. Although one can deduce this from the fact that optimal policies don't always exist (Example 1) while SBE policies do (Theorem 1), it also holds for finite MDPs and reachability goals for which optimal policies always exist.

Example 2 (SBE $\not\rightarrow$ optimal)**.** Consider Fig. 1 with $p_n := 1/10$, $q_n := 7/10$, and $r_n = 0$. The domain is an MDP. Only the first action played is relevant; thus there are in effect just two policies "first do a", which is optimal and achieves the goal with probability $7/10$, and "first do b", which is not optimal (it achieves the goal with probability $1/10$). However, both policies are SBE (since both have value 0 at s_0).

By Proposition 1 and Definition 1 we get:

Theorem 3. *Let D be a stochastic domain and G a goal. If there is a policy σ that almost-surely enforces G in D, then a policy σ' is stochastic-best effort for (D, G) if and only σ' almost-surely enforces G in D.*

This has an analogue in the non-stochastic setting: if there is a winning policy, then the best-effort policies are exactly the winning policies [Aminof *et al.*, 2020a].

4 Generalized Stochastic Planning

Finding a single policy that works in multiple "similar" domains is called "Generalized Planning". In this section we introduce and study generalized planning for stochastic domains. We restrict to stochastic domains that are bounded and have Markovian support: the latter means we can talk about an induced nondeterministic domain that does not change over time; the former means this domain does not change "in the limit" (intuitively, if the probability of taking a transition (s, a, s') goes to zero very fast, then the probability conditioned on being at s infinitely often of eventually taking the edge may go to zero, which means, intuitively, that this edge is not there "in the limit").

Definition 2. *Let D_1, D_2 be stochastic domains that are bounded and have Markovian support. We say that D_1 and D_2 are* similar *if they have the same set of fluents F, actions A, initial observation s_0, same support functions, but possibly different Pr. A* generalized planning problem *is a pair $(\mathfrak{D}, G)$ where $\mathfrak{D}$ is a set of pairwise-similar stochastic domains, and G is a goal for every $D \in \mathfrak{D}$.*

We specify G by LTL/LTL$_f$ formulas (and in some proofs also with automata), and we specify $\mathfrak{D}$ by a fully observable nondeterministic domain (FOND) $D = (F, A, s_0, \Delta)$, where $\Delta : Obs \times Act \rightarrow 2^{Obs} \setminus \{\emptyset\}$. That is, D determines the set $\mathfrak{D}$ of bounded stochastic domains $D' = (F, A, s_0, Pr)$ such that Δ is the support function of Pr.

Definition 3. *Let $(\mathfrak{D}, G)$ be a generalized planning domain. A policy is* stochastic best-effort (SBE) *(resp.* optimal*, resp.* almost-sure enforcing*) if for every domain $D \in \mathfrak{D}$, the policy is stochastic best-effort (resp. optimal, resp. almost-sure enforcing) for the planning domain (D, G).*

It is not hard to see that already in very simple cases there may not be a policy that is optimal for $\mathfrak{D}$, even though for every $D \in \mathfrak{D}$ there is an optimal policy.

Example 3. Recall the domain from Example 2 that has an optimal policy. Define a similar domain D' with the roles of a and b reversed, i.e., $p_n := 7/10, q_n := 1/10, r_n := 0$. Then the optimal policy in D' is "first do b". So, there is no policy that is optimal for both (D, G) and (D', G).

On the other hand, as we now show, already in a quite general setting there is a stochastic best-effort policy for $(\mathfrak{D}, G)$.

Theorem 4. *Let $(\mathfrak{D}, \varphi)$ be a generalized planning problem for a* LTL/LTL$_f$ *goal φ. Then:*

1. *There exists a finite-state SBE policy for $(\mathfrak{D}, \varphi)$ which can be computed from the specification of $\mathfrak{D}$ (i.e., the common Markovian support) and φ (i.e., the* LTL/LTL$_f$ *formula).*
2. *If σ is a finite-state SBE policy for (D, φ) for some $D \in \mathfrak{D}$, then σ is a SBE policy for $(\mathfrak{D}, \varphi)$. Moreover, if σ almost-surely enforces φ for some $D \in \mathfrak{D}$, then σ is almost-surely enforcing for $(\mathfrak{D}, \varphi)$.*

The algorithm for Part 1 is simple. Given the support $\Delta : Obs \times Act \rightarrow 2^{Obs} \setminus \{\emptyset\}$ and LTL/LTLf formula φ:

Step 1. Fix a similar finite MDP $D = (F, A, s_0, Pr)$ by letting $Pr(s, a)$ be, e.g., the uniform distribution over $\Delta(s, a)$: $Pr(s, a)(s') := 1/|\Delta(s, a)|$ for $s' \in \Delta(s, a)$.

Step 2. Return a finite-state optimal σ for the stochastic planning problem (D, φ). This can be computed using any standard technique, e.g., [Bianco and de Alfaro, 1995].

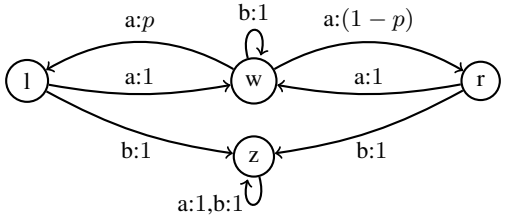

Figure 2: Stochastic domain for simulating a random walk.

Remark 1. Consider finite nondeterministic domains $D = (F, A, s_0, \Delta)$ and reachability goals $\mathsf{F}\,\varphi$. An important concept in this context are *strong-cyclic policies*, e.g., [Cimatti *et al.*, 2003]. A view for such policies is that D abstracts a single finite MDP whose exact probabilities are unknown, but whose support is known, and strong-cyclic policies almost-surely enforce $\mathsf{F}\,\varphi$. Theorem 4 (Part 2) implies that one can take a much broader view: strong-cyclic planning can be viewed as finding a policy that is almost-sure enforcing for generalized stochastic planning consisting of infinitely many, not-necessarily Markovian, similar stochastic domains.

In the special case of strong-cyclic solutions, it was noted in [Jensen *et al.*, 2001] that one can find such solutions by simply choosing a convenient instantiation and solving it for optimality. Our algorithm (Theorem 4, Part 1) extends this result in three directions: (1) LTL/LTL$_f$ instead of just reachability; (2) generalized stochastic planning instead of finite MDPs; (3) SBE instead of simply enforcing almost-surely.

Remark 2. Part 2 fails if we remove any of the three assumptions, i.e., policies being finite-state, domains being bounded, domains having Markovian support: (*i*) If σ is not assumed finite-state, consider the stochastic domain from Fig. 2. There is a non finite-state policy σ that almost-surely enforces the goal iff $p \leq 1/2$. Intuitively, the policy σ simulates a one-dimensional random walk on $\mathbb{N}_0$ starting at 1, stopping at 0, where p is the probability of decrementing; and uses the fact that the random walk almost-surely stops iff $p \leq 1/2$, see, e.g., [Bhattacharya and Waymire, 2021]. The policy σ keeps a counter $k \in \mathbb{N}_0$ as memory: if $k = 0$ and observe "l" or "r" then choose action "b", otherwise choose action "a"; decrement (resp. increment) the counter when observing "l" (resp. "r"). (*ii*) If domains are not assumed to be bounded, consider two domains D, D' instantiated from Fig. 1 with $p_n := 0$ and $r_n := 1 - q_n$. For D let $q_n := 1/3$, and for D' let $q_n := 1/3^{n+1}$. Then consider the policy that always plays a. (*iii*) If the domains are not assumed to have Markovian support, consider D_1, D_2 instantiated from Fig. 1 with $r_n := 1$. Domain D_1 is defined $p_0 := 1$ and $p_n := 0$ for $n > 0$, while D_0 is defined by $p_0 := 0$ and $p_n := 1$ for $n > 0$. Then consider the policy that immediately plays b.

4.1 Automata-theory preliminaries

We define the (synchronous) product of a deterministic automaton and a stochastic domain. This allows us to reduce finite-state policies to memoryless policies, and to work with automata acceptance conditions instead of LTL/LTL$_f$ goals.

A *deterministic automaton* is a tuple $\mathcal{M} = (\Sigma, Q, q_0, \delta)$ where Σ is the *input alphabet*, Q is a finite set of *states*, $q_0 \in Q$ is the *initial state* and $\delta : Q \times \Sigma \to Q$ is the *transition function*. A (finite or infinite) input string $u = u_0 u_1 \cdots$ determines a unique *run*, i.e., the sequence $q_0 q_1 \cdots$ of states starting with the initial state such that $\delta(q_i, u_i) = q_{i+1}$ for all $0 \leq i < |u|$. We use labeling functions $\lambda : Q \to L$ that map the automaton states Q to some set of labels: we set $L = Act$ to model the decisions taken by finite-state policies; we set $L = \{0, 1\}$ to model *reachability objectives* for deterministic finite word automata (DFAs); we set $L \subseteq \mathbb{Z}$ to model *parity objectives* for deterministic parity word automata (DPAs). A DFA $\mathcal{M}$ accepts a finite string u if $\lambda(q_i) = 1$ for the last state q_i of the run of $\mathcal{M}$ on u. A DPA $\mathcal{M}$ accepts an infinite string u if $\limsup_{i\to\infty} \lambda(q_i)$ is even, where the q_i are the states of the infinite run of $\mathcal{M}$ on u (i.e., if the largest integer that occurs infinitely often as a label of a state in the run is even).

We now define the product of a deterministic automaton and a stochastic domain. Intuitively, the product will result in a stochastic domain with the "same" distribution, i.e., the product domain $D \times \mathcal{M}$ mimics the probabilistic choices of D while also deterministically storing the state that $\mathcal{M}$ would reach given the history so far. Here is the construction. Let $D = (F, A, s_0, \mathsf{Pr})$ be a stochastic domain and let $\mathcal{M} = (\Sigma, Q, q_0, \delta)$ be a deterministic automaton with $Q = 2^B$ for some set B of Boolean variables disjoint from F (this is a technical convenience), and $\Sigma = 2^{F \cup A}$. Define the *synchronous product* $D \times \mathcal{M}$ to be the stochastic domain $(F \cup B, A, s_0 \cup q_0, \mathsf{Pr}')$ where the probability distribution Pr' is defined as follows. For a history $x = (s_0 \cup q_0, a_0, s_1 \cup q_1, a_1, \cdots, s_k \cup q_k)$ of the product, and $a \in Act$, the support of $\mathsf{Pr}'(x, a)$ is the set valuations $s \cup q$ such that (i) s is in the support of $\mathsf{Pr}((s_0, a_1, s_1, a_1, \cdots, s_k), a)$, and (ii) $q = \delta(q_k, s_k \cup a)$, and in this case, $\mathsf{Pr}'(x, a)(s \cup q) = \mathsf{Pr}((s_0, a_0, s_1, a_1, \cdots, s_k), a)$. Every labeling function $\lambda : Q \to L$ can be lifted to $D \times \mathcal{M}$ by setting $\lambda(s, \ell) := \lambda(\ell)$.

We note that because we only consider deterministic automata, policies over D and over $D \times \mathcal{M}$ are in correspondence, and the measure of events is unchanged by this correspondence. We now state this formally. We say that a history $(s_0, a_0, \cdots, s_k)$ in D *induces* a history $(s_0 \cup q_0, a_0, s_1 \cup q_1, a_1, \cdots, s_k \cup q_k)$ in $D \times \mathcal{M}$, by adding the corresponding automaton state $q_{i+1} = \delta(q_i, (s_i \cup a_i))$ for every i. Likewise, we say that a policy σ in D *induces* a policy σ' in $D \times \mathcal{M}$, and an event E of $D[\sigma]$ *induces* an event E' of $(D \times \mathcal{M})[\sigma']$. The following is a direct result of the above definitions:

Lemma 1. *For a stochastic domain D and an automaton $\mathcal{M}$, consider the product stochastic domain $D \times \mathcal{M}$. Let σ be a policy and let E be an event of $D[\sigma]$. Moreover, let σ' be the induced policy and let E be the induced event of $(D \times \mathcal{M})[\sigma']$. Then, $\mu_{D[\sigma]}(E) = \mu_{(D\times\mathcal{M})[\sigma']}(E')$.*

We recall the following fundamental fact of automata-theoretic approaches to reasoning about LTL/LTL$_f$:

Theorem 5. *[Vardi and Wolper, 1986] There is an algorithm that takes an* LTL *(resp.* LTL$_f$*) formula φ and constructs a DPA (resp. DFA) $\mathcal{M}_\varphi$ that accepts the models of φ.*

Lemma 1 and Theorem 5 immediately allow us to reduce

problems about planning for LTL/LTL$_f$ objectives, to problems about planning for parity/reachability objectives:

Corollary 1. *Let D be a stochastic domain, let σ be some policy, let φ be an* LTL *(resp.* LTL$_f$*) formula and let $\mathcal{M}_\varphi$ be the corresponding DPA (resp. DFA) with parity (resp. reachability) objective $\mathcal{C}$. Moreover, let σ' be the induced policy of $D \times \mathcal{M}_\varphi$. Then, $\mu_{D[\sigma]}(\varphi) = \mu_{(D\times\mathcal{M}_\varphi)[\sigma']}(\mathcal{C})$.*

4.2 Values transfer across similar domains

In this section we show how to compare $val_{D_1,G}(\sigma, h)$ and $val_{D_2,G}(\sigma, h)$ for similar domains D_1, D_2. We begin assuming G is a parity goal and σ is memoryless, and end assuming G is an LTL/LTL$_f$ formula and σ is finite-state.

Suppose D is a stochastic domain with Markovian support and let $\Delta : Obs \times Act \to 2^{Obs}$ be the support function. A play $\pi = (s_0, a_0, s_1, a_1, \cdots)$ in D is called *state-action fair* if for every triple (s, a, s') with $s' \in \Delta(s, a)$, if there are infinitely many i such that $s_i = s$ and $a_i = a$, then there are infinitely many i such that $s_i = s$, $a_i = a$, and $s_{i+1} = s'$. The next lemma follows from [Baier and Kwiatkowska, 1998]; intuitively, if a transition has positive probability, then the probabilitity of visiting its source and not taking the transition goes to zero as the number of visits goes to infinity:

Lemma 2 (State-action fair)**.** *Let D be a bounded stochastic domain with Markovian support, and σ a policy. Let E be the set of state-action fair plays. Then $\mu_{D[\sigma]}(E) = 1$.*

For memoryless policies and parity goals, we can relate satisfaction almost-surely to satisfaction on all state-action fair plays. The reason this is possible is that state-action fairness guarantees that every play will eventually reach a bottom strongly-connected component and that every state in this component will be visited infinitely often:

Lemma 3. *Let D be a bounded stochastic domain with Markovian support, $\mathcal{C} : Obs \to \mathbb{Z}$ be a parity objective, and $\sigma : Obs \to Act$ be a memoryless policy. Then, $\mu_{D[\sigma]}(\mathcal{C}) = 1$ iff all state-action fair plays in $D[\sigma]$ satisfy $\mathcal{C}$.*

We can now show that values transfer for parity goals; this follows from Lemma 3 using that parity conditions are closed under complement and that the satisfaction of a parity condition does not depend on finite prefixes:

Theorem 6. *Let D_1, D_2 be similar stochastic domains (that are bounded and have Markovian support). Let h be a history, $\sigma : Obs \to Act$ be a finite-state policy, and $\mathcal{C} : Obs \to \mathbb{Z}$ be a parity objective. Then $val_{D_1,\mathcal{C}}(\sigma, h) = val_{D_2,\mathcal{C}}(\sigma, h)$ for every σ-history h.*

Combining Theorem 6 with Lemma 1 and Corollary 1 we can show that values transfer for LTL/LTL$_f$ goals:

Theorem 7. *Let D_1, D_2 be similar stochastic domains. Let φ be an* LTL/LTL$_f$ *goal, and let σ be a finite-state policy. For every σ-history h, $val_{D_1,\varphi}(\sigma, h) = val_{D_2,\varphi}(\sigma, h)$.*

We are now in a position to finish the proof of Theorem 4. To see that the algorithm in Part 1 is correct, note that by Theorem 2, the finite-state policy σ produced by the algorithm is SBE for (D, φ). Since D is bounded, by Theorem 7 also σ is SBE for (D', φ) for every $D' \in \mathfrak{D}$. To see that Part 2 is correct, simply apply the definitions and Theorem 7.

5 Related Work

Variants of "best-effort" policies are studied for the non-stochastic setting in Reactive Synthesis [Aminof *et al.*, 2020a; Aminof *et al.*, 2021b; Aminof *et al.*, 2021a] and in games on graphs [Berwanger, 2007; Faella, 2009; Brenguier *et al.*, 2017].

Stochastic domains in this paper are countable and not necessarily Markovian. Memory requirements for ϵ-optimal policies for countable Markovian domains with parity objectives were studied in [Kiefer *et al.*, 2020].

Various traditional notions of plan correctness for generalized planning for infinite stochastic domains with reachability goals are studied in [Belle and Levesque, 2016], which in addition, characterizes the strong-cyclic policies for generalized planning problems specified by a finite-state nondeterministic domain (with varying initial states). In contrast, the generalized planning problem in our work consist of infinitely many stochastic domains in which the probabilities are different.

The finite domain D that specifies the set of similar domains $\mathfrak{D}$ can be viewed as an abstraction $\mathfrak{D}$. This point of view has been exploited to solve (non-stochastic) generalized planning problems in [Bonet *et al.*, 2017; Bonet *et al.*, 2020].

Acknowledgments

Partially supported by the Austrian Science Fund (FWF) P 32021; ERC Advanced Grant WhiteMech (No. 834228); EU ICT-48 2020 project TAILOR (No. 952215); PRIN project RIPER (No. 20203FFYLK); JPMorgan AI Faculty Research Award "Resilience-based Generalized Planning and Strategic Reasoning".

References

[Aminof *et al.*, 2020a] Benjamin Aminof, Giuseppe De Giacomo, Alessio Lomuscio, Aniello Murano, and Sasha Rubin. Synthesizing strategies under expected and exceptional environment behaviors. In *IJCAI*, 2020.

[Aminof *et al.*, 2020b] Benjamin Aminof, Giuseppe De Giacomo, and Sasha Rubin. Stochastic fairness and language-theoretic fairness in planning on nondeterministic domains. In *ICAPS*, 2020.

[Aminof *et al.*, 2021a] Benjamin Aminof, Giuseppe De Giacomo, Alessio Lomuscio, Aniello Murano, and Sasha Rubin. Synthesizing best-effort strategies under hierarchical environment specifications. In *KR*, 2021.

[Aminof *et al.*, 2021b] Benjamin Aminof, Giuseppe De Giacomo, and Sasha Rubin. Best-effort synthesis: Doing your best is not harder than giving up. In *IJCAI*, 2021.

[Bacchus and Kabanza, 2000] Fahiem Bacchus and Froduald Kabanza. Using temporal logics to express search control knowledge for planning. *Artif. Intell.*, 116(1-2), 2000.

[Baier and Kwiatkowska, 1998] Christel Baier and Marta Z. Kwiatkowska. On the verification of qualitative properties of probabilistic processes under fairness constraints. *Inf. Process. Lett.*, 66(2):71–79, 1998.

[Baier and McIlraith, 2006] Jorge A. Baier and Sheila A. McIlraith. Planning with first-order temporally extended goals using heuristic search. In *AAAI*, 2006.

[Baier *et al.*, 2008] Christel Baier, Joost-Pieter Katoen, and Kim Guldstrand Larsen. *Principles of Model Checking*. The MIT Press, 2008.

[Belle and Levesque, 2016] Vaishak Belle and Hector J. Levesque. Foundations for generalized planning in unbounded stochastic domains. In *KR*, 2016.

[Berwanger, 2007] Dietmar Berwanger. Admissibility in infinite games. In *STACS*, 2007.

[Bhattacharya and Waymire, 2021] Rabi Bhattacharya and Edward C Waymire. *Random Walk, Brownian Motion, and Martingales*. Springer, 2021.

[Bianco and de Alfaro, 1995] Andrea Bianco and Luca de Alfaro. Model checking of probabilistic and nondeterministic systems. In *FSTTCS*, 1995.

[Bonet *et al.*, 2010] Blai Bonet, Héctor Palacios, and Hector Geffner. Automatic derivation of finite-state machines for behavior control. In *AAAI*, 2010.

[Bonet *et al.*, 2017] Blai Bonet, Giuseppe De Giacomo, Hector Geffner, and Sasha Rubin. Generalized planning: Non-deterministic abstractions and trajectory constraints. In *IJCAI*, 2017.

[Bonet *et al.*, 2020] Blai Bonet, Giuseppe De Giacomo, Hector Geffner, Fabio Patrizi, and Sasha Rubin. High-level programming via generalized planning and LTL synthesis. In *KR*, 2020.

[Brenguier *et al.*, 2017] Romain Brenguier, Jean-François Raskin, and Ocan Sankur. Assume-admissible synthesis. *Acta Inf.*, 54(1), 2017.

[Cimatti *et al.*, 1998] Alessandro Cimatti, Marco Roveri, and Paolo Traverso. Strong planning in non-deterministic domains via model checking. In *AIPS*, pages 36–43. AAAI, 1998.

[Cimatti *et al.*, 2003] Alessandro Cimatti, Marco Pistore, Marco Roveri, and Paolo Traverso. Weak, strong, and strong cyclic planning via symbolic model checking. *Artificial Intelligence*, 1–2(147), 2003.

[Daniele *et al.*, 1999] Marco Daniele, Paolo Traverso, and Moshe Y. Vardi. Strong cyclic planning revisited. In *ECP*, 1999.

[De Giacomo and Vardi, 2013] Giuseppe De Giacomo and Moshe Y. Vardi. Linear temporal logic and linear dynamic logic on finite traces. In *IJCAI*, 2013.

[D'Ippolito *et al.*, 2018] Nicolás D'Ippolito, Natalia Rodríguez, and Sebastian Sardiña. Fully observable non-deterministic planning as assumption-based reactive synthesis. *J. Artif. Intell. Res.*, 61:593–621, 2018.

[Faella, 2009] Marco Faella. Admissible strategies in infinite games over graphs. In *MFCS*, 2009.

[Geffner and Bonet, 2013] H. Geffner and B. Bonet. *A Concise Introduction to Models and Methods for Automated Planning*. M&C, 2013.

[Geffner and Geffner, 2018] Tomas Geffner and Hector Geffner. Compact policies for fully observable non-deterministic planning as SAT. In *ICAPS*, 2018.

[Ghallab *et al.*, 2016] Malik Ghallab, Dana S. Nau, and Paolo Traverso. *Automated planning and action*. CUP, 2016.

[Haslum *et al.*, 2019] Patrik Haslum, Nir Lipovetzky, Daniele Magazzeni, and Christian Muise. *An Introduction to the Planning Domain Definition Language*. M&C, 2019.

[Hu and De Giacomo, 2011] Yuxiao Hu and Giuseppe De Giacomo. Generalized planning: Synthesizing plans that work for multiple environments. In *IJCAI*, 2011.

[Jensen *et al.*, 2001] Rune M. Jensen, Manuela M. Veloso, and Michael H. Bowling. OBDD-based optimistic and strong cyclic adversarial planning. In *ECP*, 2001.

[Kiefer *et al.*, 2020] Stefan Kiefer, Richard Mayr, Mahsa Shirmohammadi, and Patrick Totzke. Strategy complexity of parity objectives in countable mdps. In *CONCUR*, 2020.

[Kissmann and Edelkamp, 2011] Peter Kissmann and Stefan Edelkamp. Gamer, a general game playing agent. *Künst. Intell.*, 25(1):49–52, 2011.

[Mattmüller *et al.*, 2010] Robert Mattmüller, Manuela Ortlieb, Malte Helmert, and Pascal Bercher. Pattern database heuristics for fully observable nondeterministic planning. In *ICAPS*, 2010.

[Muise *et al.*, 2012] Christian J. Muise, Sheila A. McIlraith, and J. Christopher Beck. Improved non-deterministic planning by exploiting state relevance. In *ICAPS*, 2012.

[Pistore and Traverso, 2001] Marco Pistore and Paolo Traverso. Planning as model checking for extended goals in non-deterministic domains. In *IJCAI*, 2001.

[Puterman, 2005] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 2005.

[Ramírez and Sardiña, 2014] Miquel Ramírez and Sebastian Sardiña. Directed fixed-point regression-based planning for non-deterministic domains. In *ICAPS*, 2014.

[Rintanen, 2004] Jussi Rintanen. Complexity of planning with partial observability. In *ICAPS*, 2004.

[Srivastava, 2010] Siddharth Srivastava. *Foundations and Applications of Generalized Planning*. PhD thesis, University of Massachusetts Amherst, 2010.

[Vardi and Wolper, 1986] Moshe Y. Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification. In *LICS*, 1986.