A PSPACE Algorithm for Almost-Sure Rabin Objectives in Multi-Environment MDPs

Marnix Suilen[1]
Radboud University, Nijmegen, The Netherlands

Marck van der Vegt[1]
Radboud University, Nijmegen, The Netherlands

Sebastian Junges
Radboud University, Nijmegen, The Netherlands

Abstract

Markov Decision Processes (MDPs) model systems with uncertain transition dynamics. Multiple-environment MDPs (MEMDPs) extend MDPs. They intuitively reflect finite sets of MDPs that share the same state and action spaces but differ in the transition dynamics. The key objective in MEMDPs is to find a single strategy that satisfies a given objective in every associated MDP. The main result of this paper is PSPACE-completeness for almost-sure Rabin objectives in MEMDPs. This result clarifies the complexity landscape for MEMDPs and contrasts with results for the more general class of partially observable MDPs (POMDPs), where almost-sure reachability is already EXP-complete, and almost-sure Rabin objectives are undecidable.

2012 ACM Subject Classification Theory of computation → Logic and verification

Keywords and phrases Markov Decision Processes, partial observability, linear-time Objectives

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2024.40

Related Version *Full Version*: `https://arxiv.org/abs/2407.07006` [38]

Funding *Marnix Suilen*: NWO OCENW.KLEIN.187
Sebastian Junges: NWO Veni Grant ProMiSe (222.147)

1 Introduction

Markov decision processes (MDPs) are the ubiquitous model for decision-making under uncertainty [34]. An elementary question in MDPs concerns the existence of strategies that satisfy qualitative temporal properties, such as *is there a strategy such that the probability of reaching a set of target states is one*? Qualitative properties in MDPs have long been considered as pre-processing for probabilistic model checking of quantitative properties [6, 24]. Recently, however, qualitative properties have received interest in the context of shielding [1, 29], *i.e.*, the application of model-based reasoning to ensure safety in reinforcement learning [25, 39].

An often prohibitive assumption in using MDPs is that the strategy can depend on the precise state. To follow such a strategy, one must precisely observe the state of the system, *i.e.*, of an agent and its environment. The more general partially observable MDPs (POMDPs) [30] do not make this assumption. In POMDPs, a strategy cannot depend on the precise states of the system but only on the (sequence of) observed labels of visited states. As a consequence, and in contrast to MDPs, winning strategies may require memory. Indeed, the existence of strategies that satisfy qualitative objectives on MDPs is efficiently decidable in polynomial time using standard graph-algorithms [15, 14]. In contrast, in POMDPs, deciding almost-sure reachability is already EXPTIME-complete [4, 14], and the existence of strategies for a more general class of almost-sure Rabin objectives is undecidable [4].

[1] Both authors contributed equally

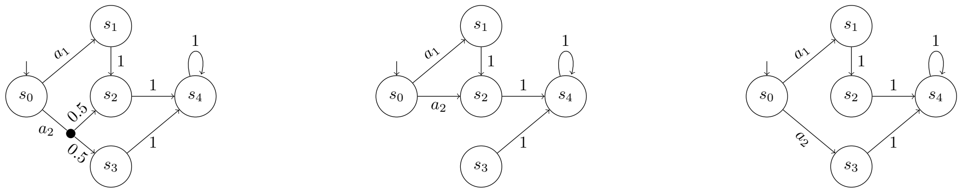

Figure 1 A MEMDP with three environments.

Multi-environment MDPs (MEMDPs) [36] model a finite set of MDPs, called *environments*, that share the state space but whose transition relation may be arbitrarily different. For any given objective, the key decision problem asks to find a single strategy that satisfies the objective in every associated MDP. MEMDPs are particularly suitable to model settings where one searches for a winning strategy that is robust to perturbations or random initialization problems. Examples of MEMDPs range from code-breaking games such as Mastermind, card games such as Free-cell, or Minesweeper, to more serious applications in robotics [17], *e.g.*, high-level planning where an artifact with unknown location must be recovered.

MEMDPs are special POMDPs [11]: An agent can observe the current state but not the transition relation determining the outcomes of its actions. This type of partial observation has an important effect: When an agent observes the next state, it may rule out that a certain MDP describes the true system. Thus, the set of MDPs that may describe the system is monotonically decreasing [11]. We call this property as *monotonic information gain*.

We illustrate MEMDPs in Fig. 1. This MEMDP consists of three environments. When playing action a_1 in state s_0, we end up in state s_1 in every environment. If we play action a_2 and observe that we end up in state s_2, we can infer that we have to either be in the left or the middle environment, while observing s_3 after a_2 rules out the middle environment. Such information cannot be lost in a MEMDP, hence *monotonic* information gain.

The most relevant results for qualitative properties on MEMDPs are by Raskin and Sankur [36], and by Van der Vegt et al. [40]. The former paper focuses on the case with only *two environments*, which we refer to as 2-MEMDPs (and more generally, k-MEMDPs). It shows, among others, that almost-sure parity can be decided in polynomial time. In 2-MEMDPs, the memory for a winning strategy is polynomial in the size of the MEMDP, while for arbitrary environments, winning strategies for almost-sure reachability may be exponential [40]. However, despite the need for exponential strategies, almost-sure *reachability* in MEMDPs is decidable in PSPACE via a recursive algorithm that exploits the aforementioned monotonic information gain [40].

Table 1 Known complexity (completeness) for MEMDPs, new results are in boldface. NL and EXP denote the classes NLOGSPACE and EXPTIME, and UD denotes UNDECIDABLE.

Semantics	Almost-sure					Possible		
Model	MDP	2-MEMDP	k-MEMDP	MEMDP	POMDP	MDP	MEMDP	POMDP
Reachability	P [14]	P [36]	**P** Cor. 49	PSPACE [40]	EXP [4, 14]	NL [14]	**NL** Thm. 6	NL [14]
Safety	P [14]	P [36]	**P** Cor. 49	**PSPACE** Thm. 41	EXP [8, 35]	P [14]	**P** Thm. 6	EXP [14]
Büchi	P [14]	P [36]	**P** Cor. 49	**PSPACE** Thm. 41	EXP [4, 14]	P [14]	**P** Thm. 6	UD [4]
Co-Büchi	P [14]	P [36]	**P** Cor. 49	**PSPACE** Thm. 41	UD [4]	P [14]	**P** Thm. 6	EXP [14]
Parity	P [14]	P [36]	**P** Cor. 49	**PSPACE** Thm. 41	UD [4]	P [14]	**P** Thm. 6	UD [4]
Rabin	P [13]	**P** Cor. 49	**P** Cor. 49	**PSPACE** Thm. 41	UD [4]	P [6]	**P** Thm. 6	UD [4]

The main result of this paper is a landscape of qualitative Rabin objectives and their subclasses in MEMDPs, see Tbl. 1. The key novelty is a PSPACE algorithm to decide the existence of strategies in MEMDPs that satisfy an almost-sure Rabin objective. The algorithm relies on two key ingredients: First, as shown in Sect. 4, for almost-sure Rabin objectives, a particular type of finite-memory strategies (with memory exponential in the number of environments) is sufficient, in contrast to POMDPs. Second, towards an algorithm, we observe that a traditional, per Rabin-pair, approach for Rabin objectives does not generalize to MEMDPs (Sect. 6.1). It does, however, generalize to what we call belief-local MEMDPs, in which one, intuitively, cannot gain any information (Sect. 6.3). Exploiting the monotonic information gain of (general) MEMDPs, we construct a recursive algorithm with polynomial stack size, inspired by [40], that solves Rabin objectives in belief-local MEMDPs (Sect. 6.4). Finally, we establish PSPACE-hardness for almost-sure safety and clarify that for possible objectives, MEMDPs can be solved as efficiently as MDPs. Proofs are in the appendix of the full version of this paper [38].

Related Work

Besides almost-sure objectives, Raskin and Sankur [36] also study *limit-sure* objectives for MEMDPs of two environments. Where almost-sure objectives require that the satisfaction probability of the objective equals one, an objective is satisfied *limit-surely* whenever for any $\epsilon > 0$, there is a strategy under which the objective is satisfied with probability at least $1 - \epsilon$. For 2-MEMDPs, limit-sure parity objectives are decidable in P [36].

Closely related to the study of almost-sure objectives in MEMDPs and POMDPs is the *value* 1 *problem* for probabilistic automata (PA). A PA can be seen as a POMDP where all states have the same observation and are thus indistinguishable. The value 1 problem is to decide whether the supremum of the acceptance probability over all words equals one. This problem is undecidable for general PA [26], but recent works have studied several subclasses of PA for which the value 1 problem is decidable. Most notably, *#-acyclic* PA [26], *structurally simple* and *simple* PA [16], and *leaktight* PA [23]. Leaktight PA are the most general of these subclasses [20]. They contain the others, and the value 1 problem is PSPACE-complete [22].

The interpretation of MEMDPs as a special POMDP is successfully used in the quantitative setting, where the goal is to find a strategy that maximizes the probability of reaching a target. Finding a strategy that maximizes the finite-horizon expected reward in MEMDPs is PSPACE-complete [9], as is also the case for the same problem in more general POMDPs [33].

Besides a special class of POMDP, MEMDPs are also a class of *robust* MDP with discrete uncertainty sets [32, 27]. In the robotics and AI communities MEMDPs are studied in that context, primarily for quantitative objectives such as maximizing discounted reward or regret minimization [37]. *Parametric* MDPs (pMDPs) are another formalism for defining MDPs with a range of transition functions [28]. Where we seek a single strategy that is winning for all environments, parameter synthesis is often about finding a single parameter instantiation (or: environment) such that all strategies are winning [18, 2]. That is, the quantifiers are reversed. A notable exception is work on quantitative properties in pMDPs by Arming et al. [3], which interprets a parametric MDP as a MEMDP and solves it as a POMDP. With the (altered) quantifier order in pMDPs, memoryless deterministic strategies are sufficient: The complexity of finding a parameter instantiation such that under all (memoryless deterministic) strategies a quantitative reachability objective is satisfied is in NP when the number of parameters is fixed and ETR-complete in the general case [41]. Determining whether a memoryless deterministic policy is robust is both ETR and co-ETR-hard [41].

Concurrent parameterized games have a similar type of partial observability as MEMDPs but lack a probabilistic transition function. The complexity of deciding reachability objectives in concurrent parameterized games is PSPACE-complete [7], equal to that of almost-sure reachability objectives in MEMDPs [40].

2 Background and Notation

Let $\mathbb{N}$ denote the natural numbers. For a set X, the powerset of X is denoted by $\mathcal{P}(X)$, and the disjoint union of two sets X, Y is denoted $X \sqcup Y$. A discrete probability distribution over a finite set X is a function $\mu\colon X \to [0,1]$ with $\sum_{x \in X} \mu(x) = 1$, the set of all discrete probability distributions over X is $\mathit{Dist}(X)$. The support of a distribution $\mu \in \mathit{Dist}(X)$ is the set of elements x with $\mu(x) > 0$ and is denoted by $\mathit{Supp}(\mu)$. We denote the uniform distribution over X by $\mathsf{unif}(X)$ and the Dirac distribution with probability 1 on x by $\mathsf{dirac}(x)$.

2.1 Markov Decision Processes

We briefly define standard (discrete-time) Markov decision processes and Markov chains.

▶ **Definition 1** (MDP). *A* Markov decision process *(MDP) is a tuple $M := \langle S, A, \iota, p \rangle$ where S is a finite set of states and $\iota \in S$ is the initial state, A is the finite set of* actions*, and $p\colon S \times A \rightharpoonup \mathit{Dist}(S)$ is the partial probabilistic* transition function*. By $A(s)$, we denote the set of enabled actions for s, which are the actions for which $p(s,a)$ is defined.*

For readability, we write $p(s,a,s')$ for $p(s,a)(s')$. A *path* in an MDP is a sequence of successive states and actions, $\pi = s_0 a_0 s_1 a_1 \ldots \in (SA)^* S$, such that $s_0 = \iota$, $a_i \in A(s)$, and $p(s_i, a_i, s_{i+1}) > 0$ for all $i \geq 0$, and we write $\overline{\pi}$ for only the sequence of states in π. The probability of following a path π in an MDP with transition function p is defined as $p(\pi) = p(s_0 a_0 \ldots) = \prod_{i=0} p(s_i, a_i, s_{i+1})$. The set of all (finite) paths on an MDP M is $\mathsf{Path}(M)$ (resp. $\mathsf{Path}_{\mathsf{fin}}(M)$). Whenever clear from the context, we omit the MDP M from these notations. We write $\mathit{first}(\pi)$ and $\mathit{last}(\pi)$ for the first and last state in a finite path, respectively, and the concatenation of two paths π_1, π_2 is written as $\pi_1 \cdot \pi_2$. The set of *reachable states* from $S' \subseteq S$ is $\mathsf{Reachable}(S') := \{s' \in S \mid \exists \pi \in \mathsf{Path}_{\mathsf{fin}}\colon \mathit{first}(\pi) \in S', \mathit{last}(\pi) = s'\}$. A state $s \in S$ is a *sink state* if $\mathsf{Reachable}(\{s\}) = \{s\}$. An MDP is acyclic if each state is a sink state or not reachable from its successor states. The *underlying graph* of an MDP is a tuple $\langle V, E \rangle$ with vertices $V := \{v_s \mid s \in S\}$ and edges $E := \{\langle v_s, v_{s'} \rangle \mid \forall s, s' \in S\colon \exists a \in A\colon p(s,a,s') > 0\}$.

A *sub-MDP* of an MDP $M = \langle S, A, \iota, p \rangle$ is a tuple $\langle S', A', \iota', p' \rangle$ with states $\emptyset \neq S' \subseteq S$, actions $\emptyset \neq A' \subseteq A$, initial state $\iota' \in S'$, and a transition function p' such that $\forall s \in S\colon \emptyset \neq A'(s) \subseteq A(s)$ and $\forall s, s' \in S', a \in A'(s)\colon \mathit{Supp}(p(s,a)) \subseteq S'$ and $p'(s,a,s') := p(s,a,s')$. An *end-component* of an MDP M is a sub-MDP where $\mathsf{Reachable}(S') = S'$. Sub-MDPs and end-components are standard notions; for details, cf. [19, 6, 36].

A *Markov chain* is an MDP where there is only one action available at every state: $\forall s \in S\colon |A(s)| = 1$. We write an MC as a tuple $C := \langle S, \iota, p \rangle$ where S is a set of states, $\iota \in S$ is the initial state, and $p\colon S \to \mathit{Dist}(S)$ is the transition function. Paths in MCs are sequences of successive states, and their underlying graph is analogously defined as for MDPs. A subset $T \subseteq S$ is strongly connected if for each pair of states $\langle s, s' \rangle \in T$ there exists a finite path π with $\mathit{first}(\pi) = s$ and $\mathit{last}(\pi) = s'$. A *strongly connected component* (SCC) is a strongly connected set of states T such that no proper superset of T is strongly connected. A *bottom SCC* (BSCC) is an SCC S' where no state $s \in S \setminus S'$ is reachable.

2.2 Strategies and Objectives

We now formally define strategies and their objectives. Strategies resolve the action choices in MDPs. A strategy is a measurable function $\sigma\colon \mathsf{Path}_{\mathsf{fin}} \to \mathit{Dist}(A)$ such that for all finite paths $\pi \in \mathsf{Path}_{\mathsf{fin}}$ we have $\mathit{Supp}(\sigma(\mathit{last}(\pi))) \subseteq A(\mathit{last}(\pi))$. A strategy is deterministic if it maps only to Dirac distributions, and it is memoryless if the action (distribution) only depends on the last state of every path. We write Σ for the set of all strategies.

A strategy σ applied to an MDP M induces an infinite-state MC $M[\sigma] = \langle S^*, \iota, p_\sigma \rangle$ such that for any path π: $p_\sigma(\overline{\pi}, \overline{\pi} \cdot s') = \sum_{a \in A} \sigma(\pi)(a) \cdot p(\mathit{last}(\pi), a, s')$. This MC has a probability space with a unique probability measure $\mathbb{P}_{M[\sigma]}$ via the cylinder construction [6, 21].

A strategy is a *finite-memory* strategy if it can be encoded by a stochastic Moore machine, also known as a finite-state controller (FSC) [31]. An FSC is a tuple $\mathcal{F} = \langle N, n_\iota, \alpha, \eta \rangle$, where N is a finite set of *memory nodes*, the *initial node* $n_\iota \in N$, $\alpha\colon S \times N \to \mathit{Dist}(A)$ is the *action mapping*, and $\eta\colon N \times A \times S \to \mathit{Dist}(N)$ is the *memory update function*. The induced MC $M[\mathcal{F}]$ of an MDP M and finite-memory strategy represented by an FSC $\mathcal{F}$ is finite and defined by the following product construction: $M[\mathcal{F}] = \langle S \times N, \langle \iota, n_\iota \rangle, p_\mathcal{F} \rangle$, where $p_\mathcal{F}(\langle s, n \rangle, \langle s', n' \rangle) = \sum_{a \in A} \alpha(s, n, a) \cdot p(s, a, s') \cdot \eta(n, a, s', n')$. A strategy is *memoryless* if its FSC representation has a single memory node, *i.e.*, $|N| = 1$.

We consider both *almost-sure* and *possible* objectives for MDPs and MCs with state space S. An *objective* Φ is a measurable subset of $P \subseteq S^\omega$. An MC C is almost-surely (or possibly) *winning* for an objective Φ iff $\mathbb{P}_C(\mathsf{Path}(C) \cap \Phi) = 1$ (or $\mathbb{P}_C(\mathsf{Path}(C) \cap \Phi) > 0$). A state s is winning whenever the MC with its initial state replaced by s is winning. We write $C \models \Phi$ and $s \models^C \Phi$ to denote that MC C and state s are winning for Φ.

▶ **Definition 2** (Winning). *An MDP M is winning for Φ if there exists a strategy $\sigma \in \Sigma$ such that the induced MC $M[\sigma]$ is winning for Φ, and the strategy σ is then also called winning.*

Like above, we denote winning in MDPs with $M \models \Phi$ or $s \models^M \Phi$, respectively. Sometimes, we explicitly add the winning strategy and write $M[\sigma] \models \Phi$ and $s \models^{M[\sigma]} \Phi$ for the MDP winning Φ under σ from its initial state or some other state s, respectively.

▶ **Definition 3** (Winning Region). *We call the set of states of an MDP (or MC) that are winning objective Φ the* winning region*, denoted as $\mathit{Win}_M(\Phi) = \{s \in S \mid s \models^M \Phi\}$.*

We define Rabin objectives. Let $C = \langle S, \iota, p \rangle$ be a MC with associated probability measure $\mathbb{P}_C$, $\pi \in \mathsf{Path}(C)$ a path, and $\mathit{Inf}(\pi) \subseteq S$ the set of states reached infinitely often along π.

▶ **Definition 4** (Rabin objective). *A Rabin objective is a set of Rabin pairs: $\Phi = \{\langle \mathfrak{B}_i, \mathfrak{C}_i \rangle \mid 1 \leq i \leq k.\ \mathfrak{C}_i \subseteq \mathfrak{B}_i \subseteq S\}$. A path $\pi \in \mathsf{Path}(C)$ wins Φ if there is a Rabin pair $\langle \mathfrak{B}_i, \mathfrak{C}_i \rangle$ in Φ where the path leaves the states in $\mathfrak{B}_i$ only finitely many times, and states in $\mathfrak{C}_i$ are visited infinitely often. The MC C wins a Rabin objective almost-surely (or possibly) if $\mathbb{P}_C(\pi \in \mathsf{Path}(C) \mid \exists \langle \mathfrak{B}_i, \mathfrak{C}_i \rangle \in \Phi\colon \mathit{Inf}(\pi) \subseteq \mathfrak{B}_i \wedge \mathit{Inf}(\pi) \cap \mathfrak{C}_i \neq \emptyset) = 1$ (or possibly when > 0).*

For MDPs, almost-sure and possible Rabin objectives can be solved in polynomial time [6, Thm. 10.127], and the strategies are memoryless deterministic [13, Thm. 4]. Other objectives, specifically reachability ($\Diamond T$), safety ($\Box T$), Büchi ($\Box\Diamond T$), co-Büchi ($\Diamond\Box T$), and parity are included in Rabin objectives [13] for a set $T \subseteq S$. App. A contains formal definitions.

3 Multi-Environment MDPs and the Problem Statement

Next, we introduce the multi-environment versions of MDPs and MCs. Intuitively, these can be seen as finite sets of MDPs and MCs that share the same states and actions.

▶ **Definition 5** (MEMDP). *A* multi-environment MDP *is a tuple* $\mathcal{M} = \langle S, A, \iota, \{p_i\}_{i \in I} \rangle$ *with* S, A, ι *as for MDPs, and* $\{p_i\}_{i \in I}$ *is a* finite set of transition functions*, where* I *are the* environment *indices. We also write* $\mathcal{M} = \{M_i\}_{i \in I}$ *as a set of MDPs, where* $M_i = \langle S, A, \iota, p_i \rangle$*.*

For a MEMDP $\mathcal{M}$ and a set $I' \subseteq I$, we define the restriction to I' as the MEMDP $\mathcal{M}_{\downarrow I'} = \langle S, A, \iota, \{p_i\}_{i \in I'} \rangle$. To change the initial state of $\mathcal{M}$, we define $\mathcal{M}^{\iota'} = \langle S, A, \iota', \{p_i\}_{i \in I} \rangle$.

A *multi-environment MC* (MEMC) is a MEMDP with $\forall s \in S \colon |A(s)| = 1$. A MEMC is a tuple $\mathcal{C} = \langle S, \iota, \{p_i\}_{i \in I} \rangle$ or equivalently a set of MCs $\mathcal{C} = \{C_i\}_{i \in I}$. A BSCC in a MEMC is a set of states $S' \subseteq S$ such that S' forms a BSCC in every MC $C_i \in \mathcal{C}$. The underlying graph of a MEMDP or MEMC is the disjoint union of the graphs of the environments.

Similarly to Def. 2 for MDPs, a strategy σ for a MEMDP $\mathcal{M}$ is winning for objective Φ if and only if the induced MEMC $\mathcal{M}[\sigma] = \{\mathcal{M}[\sigma]_i\}_{i \in I}$ is winning in all environments: $\forall i \in I \colon \mathcal{M}[\sigma]_i$ is winning for Φ. Winning regions, Def. 3, extend similarly to MEMDPs: $Win_{\mathcal{M}}(\Phi) = \{s \in S \mid s \models^{\mathcal{M}} \Phi\}$.

The central decision problem in this paper is:

> Given a MEMDP $\mathcal{M}$ and a Rabin objective Φ, is there a winning strategy for Φ in $\mathcal{M}$.

We assume MEMDPs are encoded as an explicit list of MDPs and each MDP is given by the explicit transition function. The value of the probabilities are not relevant.

We first consider possible semantics in MEMDPs, completing Tbl. 1. For POMDPs, co-Büchi objectives are known to be undecidable [4]. We show that for various objectives, deciding them in MEMDPs is equally hard as in their MDP counterparts.

▶ **Theorem 6.** *Deciding possible reachability objectives for MEMDPs is in* NL*. Deciding possible safety, Büchi, co-Büchi, parity and Rabin objectives for MEMDPs is in* P*.*

Using results on MDPs from [14], these upper bounds are tight. The main observation for membership is that a MEMDP is winning possibly objectives iff each MDP is possibly winning, due to a randomization over the individual winning strategies. We can then construct algorithms that solve each environment sequentially to answer the query on MEMDPs.

From here on, we focus exclusively on the almost-sure objectives.

▶ **Theorem 7.** *Almost-sure reach, safety, (co-)Büchi, and Rabin objectives for MEMDPs are* PSPACE*-hard.*

This theorem follows from [40], which shows that almost-sure reachability is PSPACE-complete. PSPACE-hardness of almost-sure safety can be established by minor modifications to the proof: In particular, the PSPACE-hardness proof for reachability operates on acyclic MEMDPs, where we may reverse the target and non-target states to change the objective from almost-sure reachability to safety. PSPACE-hardness of almost-sure (co-)Büchi, parity, and Rabin objectives follows via reduction from almost-sure reachability.

4 Belief-Based Strategies are Sufficient

In this section, we fix a MEMDP $\mathcal{M} = \langle S, A, \iota, \{p_i\}_{i \in I} \rangle$ with an almost-sure Rabin objective Φ, and constructively show a more refined version of the following statement.

▶ **Corollary 8.** *For a MEMDP* $\mathcal{M}$ *and an almost-sure Rabin objective* Φ*, if there exists a winning strategy for* Φ*, there also exists a winning finite-memory strategy* σ^* *such that the finite-state controller (FSC) for* σ^* *is exponential (only) in the number of environments.*

This corollary to Thm. 15 below immediately gives rise to EXP algorithms for the decision problem that simply iterate over all strategies. The particular shape of the strategy, a notion that we call *belief-based*, will be essential later to establish PSPACE algorithms.

4.1 Beliefs in MEMDPs

It is helpful to consider the strategy as a model for a decision-making agent. Then, in MEMDPs, the agent observes in which state $s \in S$ it currently is but does not have access to the environment $i \in I$. The hiding of environments gives rise to the notion of a *belief-distribution* in MEMDPs, akin to beliefs in partially observable MDPs [30]. A belief distribution in a MEMDP is a probability distribution over environments $\mu \in \mathit{Dist}(I)$ that assigns a probability to how likely the agent is operating in each environment. As we show below, we only need to consider the *belief-support*, *i.e.*, a subset of environments that keeps track of whether it is *possible* that the agent operates in those environments. From now on, we shall simply write *belief* instead of *belief-support*.

▶ **Definition 9** (Belief, belief update). *Given a finite path π, we define its* belief *as its last state together with the set of environments for which this path has positive probability:* $\mathsf{Belief}(\pi) = \langle \mathit{last}(\pi), \{i \in I \mid p_i(\pi) > 0\}\rangle$*. For path $\pi \cdot as'$, the belief can be characterized recursively by the* belief update *function* $\mathsf{BU}\colon S \times \mathcal{P}(I) \times A \times S \to S \times \mathcal{P}(I)$*. Let $\langle s, J\rangle = \mathsf{Belief}(\pi)$, then: $\langle s', J'\rangle = \mathsf{BU}(\langle s, J\rangle, a, s') := \mathsf{Belief}(\pi \cdot as')$, where $J' = \{j \in J \mid p_j(s, a, s') > 0\}$. We also liberally write $\mathsf{BU}(\langle s, J\rangle, a)$ for the set of beliefs $\langle s', J'\rangle$ that are possible from $\langle s, J\rangle$ via action a, and define the two projection functions: $\mathit{St}(\langle s, J\rangle) = s$ and $\mathit{Env}(\langle s, J\rangle) = J$.*

Key to MEMDPs is the notion of *revealing transitions* [36]. A revealing transition is a tuple $\langle s, a, s'\rangle$ such that there exist two environments $i, i' \in I, i \neq i'$ with $p_i(s, a, s') > 0$ and $p_{i'}(s, a, s') = 0$. Intuitively, a transition is revealing whenever observing this transition reduces the belief over environments the agent is currently in, since we observed a transition that is not possible in one or more environments. From this notion of revealing transitions immediately follows the property of monotonic information gain in MEMDPs.

▶ **Corollary 10.** *Let $\pi \cdot as'$ be a finite path. Then: (1) $\mathit{Env}(\mathsf{Belief}(\pi \cdot as')) \subseteq \mathit{Env}(\mathsf{Belief}(\pi))$, and (2) If there are environments $j, j' \in \mathit{Env}(\mathsf{Belief}(\pi))$ with $p_j(\mathit{last}(\pi), a, s') > 0 = p_{j'}(\mathit{last}(\pi), a, s'))$, then $j' \notin \mathit{Env}(\mathsf{Belief}(\pi \cdot as'))$ and thus $\mathit{Env}(\mathsf{Belief}(\pi \cdot as')) \subset \mathit{Env}(\mathsf{Belief}(\pi))$.*

Key to our analysis of MEMDPs is the notion of *belief-based* strategies.

▶ **Definition 11** (Belief-based strategy). *A strategy σ is* belief-based *when for all finite paths π, π' such that $\mathsf{Belief}(\pi) = \mathsf{Belief}(\pi')$ implies $\sigma(\pi) = \sigma(\pi')$. Belief-based strategies can also be written as a function $\sigma\colon S \times \mathcal{P}(I) \to \mathit{Dist}(A)$.*

Belief-based strategies are a form of finite-memory strategies and are representable by FSCs.

▶ **Lemma 12.** *A belief-based strategy $\sigma\colon S \times \mathcal{P}(I) \to \mathit{Dist}(A)$ for a MEMDP $\mathcal{M}$ with states S and actions A can be represented by an FSC.*

Similar to how states may be winning, beliefs can also be winning.

▶ **Definition 13** (Winning belief). *We call the belief $\langle s, J\rangle$* winning for objective Φ in $\mathcal{M}$*, written as $\langle s, J\rangle \models^{\mathcal{M}} \Phi$, if there exists a strategy $\sigma\colon \mathsf{Path}_{\mathit{fin}} \to \mathit{Dist}(A)$ such that for every environment $j \in J$, the induced MC is winning. That is, $\forall j \in J\colon \mathcal{M}_j[\sigma] \models \Phi$.*

A MEMDP $\mathcal{M}$ is winning, $\mathcal{M} \models \Phi$, iff the initial belief $\langle \iota, I \rangle$ is winning. We can extend the notion of a winning region to beliefs. The (belief) winning region of a MEMDP $\mathcal{M}$ is $Win_{\mathcal{M}}(\Phi) = \{\langle s, J \rangle \in S \times \mathcal{P}(I) \mid \langle s, J \rangle \models^{\mathcal{M}} \Phi\}$. The notion of winning beliefs has been used before in the context of POMDPs with other almost-sure objectives [10].

The induced MEMC of a MEMDP and FSC conservatively extends the standard product construction between an MDP and FSC to be applied to each transition function $\{p_i\}_{i \in I}$ individually. In that MEMC, the objective must be lifted to the new state space $S \times N$.

▶ **Definition 14** (Lifted Rabin objective). *For Rabin objective* $\Phi = \{\langle \mathfrak{B}_i, \mathfrak{C}_i \rangle \mid 1 \leq i \leq k.\ \mathfrak{C}_i \subseteq \mathfrak{B}_i \subseteq S\}$ *on* S*, the* lifted Rabin objective *to* $S \times N$ *is* $\hat{\Phi} := \{\langle \mathfrak{C}_i \times N, \mathfrak{B}_i \times N \rangle \mid 1 \leq i \leq k\}$.

When clear from the context, we implicitly apply this lifting where needed.

4.2 Constructing Winning Belief-Based Strategies

We are now ready to state the main theorem of this section.

▶ **Theorem 15.** *For MEMDP* $\mathcal{M}$ *and Rabin objective* Φ*, there exists a winning strategy* σ *for* Φ *iff there exists a belief-based strategy* σ^* *that is winning for* Φ.

The remainder of this section is dedicated to the necessary ingredients to prove Thm. 15.

▶ **Definition 16** (Allowed actions). *The set of* allowed actions *for a winning belief* $\langle s, J \rangle$ *is* $\mathsf{Allow}(\langle s, J \rangle) := \{a \in A(s) \mid \forall \langle s', J' \rangle \in \mathsf{BU}(\langle s, J \rangle, a) \colon \langle s', J' \rangle \models^{\mathcal{M}} \Phi\}$.

That is, an action at a winning belief is allowed if all possible resulting successor beliefs are still winning. Using allowed actions we define the belief-based strategy $\sigma_{\mathsf{Allow}} \colon S \times \mathcal{P}(I) \to Dist(A)$:

$$\sigma_{\mathsf{Allow}}(\langle s, J \rangle) := \begin{cases} \mathsf{unif}(\mathsf{Allow}(\langle s, J \rangle)) & \text{if } \mathsf{Allow}(\langle s, J \rangle) \neq \emptyset, \\ \mathsf{unif}(A(s)) & \text{otherwise.} \end{cases}$$

The strategy σ_{Allow} randomizes uniformly over all allowed actions when the successor beliefs are still winning and over all actions when the belief cannot be winning. We now sketch how to use σ_{Allow} to construct a winning belief-based strategy. See App. C for the details.

When playing σ_{Allow}, the induced MEMC $\mathcal{M}[\sigma_{\mathsf{Allow}}]$ will almost-surely end up in a BSCC. Given belief $\langle s, J \rangle$, we compute all environments $j \in J$ for which $\mathcal{M}_j[\sigma_{\mathsf{Allow}}]$ is in a BSCC S_{B}: $\mathcal{J}_{\langle s, J \rangle} := \{j \in J \mid \exists S_{\mathsf{B}} \subseteq S \times \mathcal{P}(I) \colon S_{\mathsf{B}} \text{ is a BSCC in MC } \mathcal{M}[\sigma_{\mathsf{Allow}}]_j \wedge \langle s, J \rangle \in S_{\mathsf{B}}\}$.

As a consequence, since σ_{Allow} is a belief-based strategy, every BSCC of the MEMC $\mathcal{M}[\sigma_{\mathsf{Allow}}]$ has a fixed set of environments that cannot change anymore. As σ_{Allow} uniformly randomizes over all allowed actions, and it remains possible to win, there has to exist a strategy $\sigma_{\mathsf{BSCC}} \colon S_{\mathsf{B}} \to Dist(A)$ that is a sub-strategy of σ_{Allow}, *i.e.*, it only plays a subset of actions that are also played by σ_{Allow}. We construct appropriate sub-MDPs to compute σ_{BSCC} for a belief $\langle s, J \rangle$ that is almost-surely winning for Φ. Using σ_{Allow} and σ_{BSCC}, we construct the following belief-based strategy and show it is indeed winning. The other direction follows since beliefs are based on paths, which proves Thm. 15.

$$\sigma^*(\langle s, J \rangle) := \begin{cases} \sigma_{\mathsf{Allow}}(\langle s, J \rangle) & \text{if } \mathcal{J}_{\langle s, J \rangle} = \emptyset, \\ \sigma_{\mathsf{BSCC}}(\langle s, J \rangle) & \text{otherwise.} \end{cases}$$

Thm. 15 shows that belief-based strategies are sufficient for almost-sure Rabin objectives in MEMDPs, which is not true on more general POMDPs [12]. Key is the monotonic information gain in MEMDPs, a property that POMDPs do not have in general [11].

▶ Remark. For the remainder of this paper, we assume all strategies are finite-memory strategies, and the induced (ME)MCs are defined via the product construction from Sect. 2.2.

5 Explicitly Adding Belief to MEMDPs

Above, we showed that it is sufficient for a strategy to reason over the beliefs. Now, we show how to add beliefs to the MEMDP, yielding a *belief observation MDP* (BOMDP). We discuss their construction (Sect. 5.1) and then algorithms on BOMDPs for reachability (Sect. 5.2) and so-called safe Büchi objectives (Sect. 5.3). We discuss Rabin objectives in Sect. 6.

5.1 Belief-Observation MDPs

We create a product construction between the MEMDP and the beliefs $\mathcal{P}(I)$ such that the beliefs of the MEMDP $\mathcal{M}$ are directly encoded in the state space:

▶ **Definition 17** (BOMDP). *The* belief observation MDP *(BOMDP) of MEMDP* $\mathcal{M} = \langle S, A, \iota, \{p_i\}_{i\in I}\rangle$ *is a MEMDP* $\mathcal{B}_\mathcal{M} = \langle S', A, \iota', \{p'_i\}_{i\in i}\rangle$ *with states* $S' = S \times \mathcal{P}(I)$*, initial state* $\iota' = \langle \iota, I\rangle$*, and partial transition functions that are defined when* $a \in A(s)$ *such that*

$$p'_j(\langle s, J\rangle, a, \langle s', J'\rangle) = \begin{cases} p_j(s,a,s') & \textit{if } \langle s', J'\rangle = \mathsf{BU}(\langle s, J\rangle, a, s') \wedge j \in J, \\ 0 & \textit{otherwise.} \end{cases}$$

BOMDPs are special MEMDPs; hence, all definitions for MEMDPs apply to BOMDPs. Due to the product construction, a belief-support-based strategy for $\mathcal{M}$ can be turned into a memoryless strategy for $\mathcal{B}_\mathcal{M}$, and vice versa. In BOMDPs, the belief J is already part of the state, so we simplify the satisfaction notation to $\langle s, J\rangle \models^{\mathcal{B}_\mathcal{M}} \Phi$ instead of $\langle\langle s, J\rangle, J\rangle \models^{\mathcal{B}_\mathcal{M}} \Phi$.

▶ **Definition 18** (Lifted strategy). *Given MEMDP* $\mathcal{M}$ *and a belief-based strategy* σ*. The* lifted memoryless strategy $\hat{\sigma}\colon (S \times \mathcal{P}(I)) \to \mathit{Dist}(A)$ *on* $\mathcal{B}_\mathcal{M}$ *is* $\hat{\sigma}(s, J) := \sigma(\langle s, J\rangle)$*.*

This lifting ensures that belief-based strategies and their liftings to BOMDPs coincide.

▶ **Lemma 19.** *Given a MEMDP* $\mathcal{M}$*, its BOMDP* $\mathcal{B}_\mathcal{M}$*, a belief-based strategy* σ *for* $\mathcal{M}$ *and its lifted strategy* $\hat{\sigma}$ *for* $\mathcal{B}_\mathcal{M}$*, we have that the two induced MEMCs coincide:* $\mathcal{M}[\sigma] = \mathcal{B}_\mathcal{M}[\hat{\sigma}]$*.*

Consequently, satisfaction of objectives is preserved by the transformation.

▶ **Theorem 20.** *Let* $\mathcal{M}$ *be a MEMDP with state space* S *and* Φ *a Rabin objective. Let* $\hat{\Phi}$ *be the lifted Rabin objective to* $S \times \mathcal{P}(I)$ *by Def. 14. A belief-based strategy* σ *for* $\mathcal{M}$ *is winning the Rabin objective* Φ *iff the lifted strategy* $\hat{\sigma}$ *is winning the lifted objective* $\hat{\Phi}$ *for* $\mathcal{B}_\mathcal{M}$*:* $\forall \sigma\colon \langle s, J\rangle \models^{\mathcal{M}[\sigma]} \Phi \Leftrightarrow \langle s, J\rangle \models^{\mathcal{B}_\mathcal{M}[\hat{\sigma}]} \hat{\Phi}$*.*

As a result of Thm. 20, we will implicitly lift strategies and objectives.

5.2 An Algorithm for Reachability in BOMDPs

In this subsection, we establish an algorithm for computing the winning region for reachability objectives in a BOMDP. The winning region of a BOMDP $\mathcal{B}_\mathcal{M}$ is precisely the set of winning beliefs of its MEMDP $\mathcal{M}$: $\mathit{Win}_{\mathcal{B}_\mathcal{M}}(\Phi) = \{\langle s, J\rangle \in S \times \mathcal{P}(I) \mid \langle s, J\rangle \models^\mathcal{M} \Phi\}$. The algorithm specializes a similar fixed-point computation for POMDPs [12] to BOMDPs.

Alg. 1 computes these winning regions. It relies on a state-remove operation defined below. Intuitively, the algorithm iteratively removes losing states, which does not affect the winning region until all states that remain in $\mathcal{B}_\mathcal{M}$ are winning.

Removing state s from a BOMDP removes the state and disables outgoing action from any state where that action that could reach s with positive probability. This operation thus also removes any action and its transitions that could reach the designated state.

Algorithm 1 Reachability algorithm for a BOMDP $\mathcal{B}_\mathcal{M}$ of MEMDP $\mathcal{M}$.

function $\textsc{Reach}$(BOMDP $\mathcal{B}_\mathcal{M}$, $T \subseteq S$)
  do
    for $i \in I$ **do**
      $S_i \leftarrow \{\langle s, J\rangle \in S \times \mathcal{P}(I) \mid i \in J\}$
      for $\langle s, J\rangle \in S_i \setminus \mathit{Win}_{\mathcal{B}_{\mathcal{M}_i}}(\Diamond T)$ **do** ▷ Iterate over all losing states
        $\mathcal{B}_\mathcal{M} \leftarrow \mathit{StateRemove}(\mathcal{B}_\mathcal{M}, \langle s, J\rangle)$ ▷ See Def. 21
  while $\bigwedge_{i\in I} S_i \neq \mathit{Win}_{\mathcal{B}_{\mathcal{M}_i}}(\Diamond T)$ ▷ Check if stable
  return $S^{\mathcal{B}_\mathcal{M}}$

▶ **Definition 21** (State removal). *Let $\mathcal{B}_\mathcal{M} = \langle S \times \mathcal{P}(I), A, \iota, \{p_i\}_{i\in I}\rangle$ be a BOMDP, and $\perp \notin S \times \mathcal{P}(I)$ a sink state. The BOMDP $\mathit{StateRemove}(\mathcal{B}_\mathcal{M}, \langle s, J\rangle)$ for $\mathcal{B}_\mathcal{M}$ and state $\langle s, J\rangle \in S \times \mathcal{P}(I)$ is given by $\langle \{\perp\} \cup S \times \mathcal{P}(I) \setminus \{\langle s, J\rangle\}, A, \iota', \{p'_i\}_{i\in I}\rangle$, where $\iota' = \perp$ if $\langle s, J\rangle = \iota$, and $\iota' = \iota$ otherwise, and for all states $\langle s', J'\rangle \neq \langle s, J\rangle$ and environments $i \in I$ we have*

$$p'_i(\langle s', J'\rangle, a) = \begin{cases} p_i(\langle s', J'\rangle, a) & \text{if } \langle s, J\rangle \notin \mathit{Supp}(p_i(\langle s', J'\rangle, a)), \\ \mathsf{dirac}(\perp) & \text{if } \langle s, J\rangle \in \mathit{Supp}(p_i(\langle s', J'\rangle, a)). \end{cases}$$

The main results in this section are the correctness and the complexity of Alg. 1:

▶ **Theorem 22.** *For BOMDP $\mathcal{B}_\mathcal{M}$ and targets T: $\mathit{Win}_{\mathcal{B}_\mathcal{M}}(\Diamond T) = \textsc{Reach}(\mathcal{B}_\mathcal{M}, T)$ in Alg. 1.*

Towards a proof, the notions of losing states and strategies as defined for MDPs also apply to BOMDP states and strategies. For BOMDPs, we additionally define *losing actions* as state-action pairs that lead with positive probability to a losing state. It follows that a BOMDP state is losing iff every action from that state is losing, and a single environment where a BOMDP state is losing suffices as a witness that the state is losing in the BOMDP (see the App. D). Finally, the following lemma is the key ingredient to the main theorem.

▶ **Lemma 23.** *Removing losing states from $\mathcal{B}_\mathcal{M}$ does not affect the winning region,* i.e., *$\langle s, J\rangle \notin \mathit{Win}_{\mathcal{B}_\mathcal{M}}(\Diamond T)$ implies $\mathit{Win}_{\mathit{StateRemove}(\mathcal{B}_\mathcal{M}, \langle s,J\rangle)}(\Diamond T) = \mathit{Win}_{\mathcal{B}_\mathcal{M}}(\Diamond T)$.*

▶ **Lemma 24.** *Alg. 1 takes polynomial time in the size of $\mathcal{B}_\mathcal{M}$.*

5.3 Safe Büchi in BOMDPs

In this section, we consider winning regions for *safe Büchi objectives* of the form $\Box\mathfrak{B} \wedge \Box\Diamond\mathfrak{C}$, where $\mathfrak{C} \subseteq \mathfrak{B} \subseteq S$. The condition $\mathfrak{C} \subseteq \mathfrak{B}$ is convenient but does not restrict the expressivity. These objectives are essential for our Rabin algorithm in Sect. 6. The main result is:

▶ **Theorem 25.** *For BOMDP $\mathcal{B}_\mathcal{M}$, $\mathit{Win}_{\mathcal{B}_\mathcal{M}}(\Box\mathfrak{B} \wedge \Box\Diamond\mathfrak{C})$ is computable in polynomial time.*

We provide the main ingredients for the proof below. We first consider arbitrary MEMDPs.

▶ **Definition 26** (State restricted (ME)MDP). *Let $M = \langle S, A, \iota, p\rangle$ be an MDP and $S' \subseteq S$ a set of states. The MDP $M_{\Box S'} := \langle S' \cup \{\perp\}, A, \iota', p'\rangle$ is M* restricted to S', *with $\perp$ a sink state, $\iota' = \iota$ if $\iota \in S'$ and $\perp$ otherwise, and for $s \in S'$, $a \in A(s)$ and $s' \in S' \cup \{\perp\}$, we define:*

$$p'(s, a, s') := \begin{cases} \sum_{s'' \in S \setminus S'} p(s, a, s'') & \text{if } s' = \perp, \\ p(s, a, s') & \text{otherwise.} \end{cases}$$

This definition conservatively extends to MEMDPs per environment i: $(\mathcal{M}_{\Box S'})_i = (\mathcal{M}_i)_{\Box S'}$.

Figure 2 Example of a BOMDP fragment with Rabin objective $\Phi = \{\langle\{s_1\},\{s_1\}\rangle, \langle\{s_2\},\{s_2\}\rangle\}$.

The winning regions of a MEMDP $\mathcal{M}$ and $\mathcal{M}_{\Box\mathfrak{B}}$ coincide as, intuitively, winning strategies must remain in $\mathfrak{B}$, thus removing other states does not affect the winning region.

▶ **Lemma 27.** *The winning regions for $\Box\mathfrak{B} \wedge \Box\Diamond\mathfrak{C}$ with $\mathfrak{C} \subseteq \mathfrak{B}$ in $\mathcal{M}_{\Box\mathfrak{B}}$ and $\mathcal{M}$ coincide.*

Satisfying the Büchi objective $\Box\Diamond\mathfrak{C}$ inside $\mathcal{M}_{\Box\mathfrak{B}}$ implies satisfying the safety condition, thus:

▶ **Lemma 28.** *The winning regions for $\Box\mathfrak{B} \wedge \Box\Diamond\mathfrak{C}$ with $\mathfrak{C} \subseteq \mathfrak{B}$ and $\Box\Diamond\mathfrak{C}$ in $\mathcal{M}_{\Box\mathfrak{B}}$ coincide.*

We can lift these lemmas to the BOMDP associated with a MEMDP.

▶ **Lemma 29.** *The winning regions for $\Box\mathfrak{B} \wedge \Box\Diamond\mathfrak{C}$ with $\mathfrak{C} \subseteq \mathfrak{B}$ in $\mathcal{B}_{\mathcal{M}}$ and $\mathcal{B}_{(\mathcal{M}_{\Box\mathfrak{B}})}$ coincide.*

Almost-sure Büchi objectives can be reduced to almost-sure reachability objectives using a construction similar to the one in [5], see the proof in the App. D for details.

▶ **Lemma 30.** *Büchi in BOMDPs is decidable in polynomial time.*

Now, to prove Thm. 25, $Win_{\mathcal{B}_{\mathcal{M}}}(\Box\mathfrak{B} \wedge \Box\Diamond\mathfrak{C})$ is computable as Büchi objective on a polynomially larger MEMDP (Lem. 29) in polynomial time (Lem. 30).

6 A Recursive PSPACE Algorithm for Rabin Objectives

We now show how to exploit the structure of BOMDPs to arrive at our PSPACE algorithm for Rabin objectives in MEMDPs. We first discuss the *non-local* behavior of Rabin objectives, and in particular, why the standard approach for almost-sure Rabin objectives for MDPs fails on BOMDPs. Then, in Sect. 6.2, we introduce *J-local MEMDPs*, which are MEMDPs where the belief J does not change. These J-local MEMDPs also occur as fragments of the BOMDPs. In J-local MEMDPs, whenever a transition is made that would cause a belief update to a strict subset of J, we transition to dedicated sink states, which we refer to as *frontier states*. These frontier states reflect transitioning into a different fragment of the BOMDP, from which all previously accessed BOMDP states are unreachable due to the monotonicity of the belief update operator. Next, in Sect. 6.3, we present an algorithm for efficiently computing the winning region of Rabin objectives on J-local MEMDPs. Finally, in Sect. 6.4, we prove that frontier states can be summarized as being either winning or losing, ultimately leading to a PSPACE algorithm for deciding Rabin objectives in MEMDPs.

6.1 Non-Local Behavior of Rabin Objectives

The traditional approach for checking almost-sure Rabin objectives on MDPs, see *e.g.* [6], computes for each state $s \in S$, whether there is a strategy that *immediately* satisfies a Rabin pair $\Phi_i = \langle\mathfrak{B}_i, \mathfrak{C}_i\rangle$, *i.e.*, satisfying $\Box\mathfrak{B}_i \wedge \Box\Diamond\mathfrak{C}_i$, and is a stronger condition. A state satisfies the Rabin condition Φ iff it almost-surely reaches the set of immediately winning states (the *win set*). The example below illustrates why this approach fails to generalize to MEMDPs.

▶ **Example 31.** In Fig. 2, we see a BOMDP for which the "MDP approach" does *not* work. First, note that the only strategy that always plays a is winning in every state. Now, consider the algorithm and the first Rabin pair $\Phi_1 = \langle\{s_1\},\{s_1\}\rangle$. State $\langle s_2, \{2\}\rangle$ does not satisfy

$\Box\{s_1\} \wedge \Box\Diamond\{s_1\}$. State $\langle s_1, \{1,2\}\rangle$ also does not belong to the win set, as in $\mathcal{M}_2$ there is a $1/2$ probability of reaching the sink state $\langle s_2, \{2\}\rangle$. For the second Rabin pair, (only) state $\langle s_2, \{2\}\rangle$ is immediately winning. Thus, the win set is the singleton set containing $\langle s_2, \{2\}\rangle$. From the initial state $\langle s_1, \{1,2\}\rangle$, it is not possible to almost-surely reach the state $\langle s_2, \{2\}\rangle$, due to $\mathcal{M}_1$. Therefore, a straightforward adaption of the traditional algorithm for MDPs would yield that the initial state is losing.

The difficulty in the example above lies in the fact that in the different environments, a different Rabin pair is satisfied. However, taking the self-loop in s_1 does not update the belief and it remains unclear whether we will eventually satisfy Φ_1 or Φ_2.

6.2 Local View on BOMDPs

We formalize J-local MEMDPs, that transition into frontier states if the belief updates.

▶ **Definition 32** (J-local MEMDPs). *Given a MEMDP $\mathcal{M} = \langle S, A, \iota, \{p_i\}_{i\in I}\rangle$, the* J-local MEMDP $\mathcal{M}\{J\} = \langle S \sqcup F, A, \{p'_j\}_{j\in J}, \iota\rangle$ *is a MEMDP, with as state space the disjoint union of the (original) states S and the* frontier states $F := S \times A \times S$. *The transition functions $\{p'_j \colon S \sqcup F \times A \rightharpoonup Dist(S \sqcup F)\}_{j\in J}$ are defined s.t. (1) $p'_j(f, a, f) = 1$ for all $f \in F$, (2) $p'_j(s, a)$ is undefined if $p_j(s, a)$ is undefined, and (3) for every state $s \in S$ and $a \in A(s)$, we define $p'_j(s, a, \langle s, a, s'\rangle) = p_j(s, a, s')$ if $\mathsf{BU}(\langle s, J\rangle, a, s') \neq \langle s', J\rangle$ and $p'_j(s, a, s') = p_j(s, a, s')$ otherwise.*

By definition of the transition functions $\{p'_j\}_{j\in J}$ of a J-local MEMDP $\mathcal{M}\{J\}$, all environments of $\mathcal{M}\{J\}$ share the same underlying graph within the states of S. Transitions to the frontiers may, however, differ (made formal in App. E). As both $\mathcal{M}$ and $\mathcal{M}\{J\}$ have states in S, a Rabin objective Φ can readily be applied to both. To give meaning to the frontier states F in $\mathcal{M}\{J\}$, we introduce *localized Rabin objectives*:

▶ **Definition 33** (Localized Rabin objective, winning frontier). *Given Rabin objective $\Phi = \{\langle \mathfrak{B}_i, \mathfrak{C}_i\rangle \mid 1 \le i \le k.\ \mathfrak{C}_i \subseteq \mathfrak{B}_i \subseteq S\}$ and some subset of frontier state $WF \subseteq F$, the* localized Rabin objective *for J-local MEMDP $\mathcal{M}\{J\}$ is $\Phi^{Loc}(WF) := \{\langle \mathfrak{B}_i \cup WF, \mathfrak{C}_i \cup WF\rangle \mid 1 \le i \le k\}$. We call WF the* winning frontier*, as any path that reaches a state in WF is winning.*

6.3 An Algorithm for Localized Rabin Objectives

Below, we present an algorithm to compute the winning region of a localized Rabin objective on a J-local MEMDP, using some auxiliary definitions on winning in a J-local MEMDPs.

▶ **Definition 34** (Immediately winning Rabin pair/state). *A J-local MEMDP state $s \in S \sqcup F$ has an* immediately winning Rabin pair $\Phi_i = \langle \mathfrak{B}_i, \mathfrak{C}_i\rangle$ *when $s \models^{\mathcal{M}\{J\}} \Box\mathfrak{B}_i \wedge \Box\Diamond\mathfrak{C}_i$. A state $s \in S \sqcup F$ is* immediately winning *if it has an immediately winning Rabin pair.*

Immediately winning states are, in particular, also winning states (see Lem. 59, App. E). It is natural also to consider specialized winning regions for just immediately winning states:

▶ **Definition 35.** *The* Rabin win set $W_{\Phi^{Loc}}$ *is $\{s \in S \sqcup F \mid s$ is immediately winning $\}$.*

The crux of our algorithm is that in J-local MEMDPs, as in MDPs but unlike in BOMDPs, winning a Rabin objective is equivalent to almost-surely reaching the Rabin win set.

▶ **Lemma 36.** *A state s in a J-local MEMDP is winning iff it can almost-surely reach $W_{\Phi^{Loc}}$.*

Algorithm 2 Local Rabin Algorithm.

```
1: function RABIN(Local MEMDP 𝓛 = 𝓜{J}, WF, Φ = {⟨𝔅_1, ℭ_1⟩, ··· , ⟨𝔅_n, ℭ_n⟩} )
2:     S_win ← ∅
3:     for 1 ≤ i ≤ n do
4:         𝔅'_i ← 𝔅_i ∪ WF  ;  ℭ'_i ← ℭ_i ∪ WF
5:         S_win ← S_win ∪ Win_𝓛(□𝔅'_i ∧ □◇ℭ'_i)          ▷ See Thm. 25
6:     return Win_𝓛(◇S_win)
```

We sketch the proof ingredients later. We first introduce Alg. 2, which lifts the MDP approach (Sect. 6.1) to J-local MEMDPs. The set S_{win} on line 2 stores states for which an immediately winning Rabin pair has been found. For each Rabin pair Φ_i, the algorithm computes the localized Rabin pair Φ_i^{Loc}. Next, in line 5, it compute the winning region $Win_{\mathcal{L}}(\Box\mathfrak{B}'_i \wedge \Box\Diamond\mathfrak{C}'_i)$ using the approach described in Sect. 5.3. These are exactly the states that have Φ_i^{Loc} as an immediately winning Rabin pair, *i.e.*, they constitute the win set S_{win}. Finally, the algorithm outputs the winning region by computing states that almost-surely reach S_{win} using Alg. 1.

▶ **Theorem 37.** *Alg. 2 yields winning regions for local MEMDPs and localized Rabin objectives.*

The remainder of this subsection discusses the ingredients for proving Lem. 36 and the theorem above. Therefore, we consider the induced Markov chain C of environment j under any strategy, *i.e.*, $C = \mathcal{M}\{J\}[\sigma]_j$. In any state that is in a BSCC of C, we notice that the reachable states *in any environment* are contained by the BSCC and the frontier states. Furthermore, we observe that in any environment, either the BSCCs in those states are the original BSCC or are (trivial) BSCCs in the frontier. Formal statements are given in App. E. The next lemma shows that states that are (under a winning strategy and in some environment) in a BSCC are immediately winning with some Rabin pair. The main challenge is that this BSCC may not be a BSCC in every environment. Using the observations above, if the states do not constitute a BSCC, they will almost surely reach (winning) frontier states, which allows us to derive the following formal statement:

▶ **Lemma 38.** *Given a J-local MEMDP $\mathcal{M}\{J\}$ and a winning strategy σ. Every state that is in a BSCC $S_{\mathsf{B}j}$ of $\mathcal{M}\{J\}[\sigma]_j$ of some environment $j \in J$, is in $W_{\Phi^{Loc}}$.*

With this statement, we can now prove Lem. 36 as under any winning strategy, we almost-surely end up in BSCCs. We return to the proof of the main theorem about the correctness of Alg. 2. First, we observe that we correctly identify the immediately winning states.

▶ **Lemma 39.** *Alg. 2 computes the set of states that are immediately winning, $W_{\Phi^{Loc}}$.*

Lems. 36 and 39 together prove Thm. 37. Finally, we remark:

▶ **Lemma 40.** *Alg. 2 is a polynomial time algorithm.*

6.4 Recursive Computation of Winning Regions

We now detail how to combine the local computations of winning regions towards a global winning region. Furthermore, we show that to obtain the winning region at the root (*i.e.*, I-local), we can forget about the winning regions below and, consequently, present a recursive approach (akin to [40]) to decide almost-sure Rabin objectives for MEMDPs in PSPACE.

▶ **Theorem 41.** *Winning almost-sure Rabin objectives in MEMDPs is decidable in* PSPACE.

Algorithm 3 Generic recursive algorithm for MEMDPs.

function $\textsc{Check}(\text{MEMDP } \mathcal{M} = \langle S, A, \iota, \{p_i\}_{i \in I}\rangle, \Phi\ )$
  $\mathcal{L} \leftarrow \mathcal{M}\{I\}$
  $RF \leftarrow \mathsf{Reachable}(S^{\mathcal{L}}) \cap F^{\mathcal{L}}$ ▷ Compute the reachable frontier states
  $WF \leftarrow \{\langle s, a, s'\rangle \in RF \mid \langle s', J'\rangle = \mathsf{BU}(\langle s, J\rangle, a, s') \wedge \textsc{Check}(\mathcal{M}^{s'}{\downarrow_{J'}}, \Phi)\}$
  return $\iota \in \textsc{Rabin}(\mathcal{L}, WF, \Phi)$ ▷ Compute winning set with winning frontier

In the remainder, we show this by providing a recursive algorithm and proving its correctness. An important construction is to project the winning region into a particular set of beliefs.

▶ **Definition 42** (Belief-restricted winning regions). *For a Rabin objective Φ, we define the following restrictions of the winning region: (1) $Win_{\mathcal{M}}(\Phi)_J := Win_{\mathcal{M}}(\Phi) \cap (S \times \{J\})$, (2) $Win_{\mathcal{M}}(\Phi)_{\subset J} := \bigcup_{J' \subset J} Win_{\mathcal{M}}(\Phi)_{J'}$, and (3) $Win_{\mathcal{M}}(\Phi)_{\subseteq J} := Win_{\mathcal{M}}(\Phi)_{\subset J} \cup Win_{\mathcal{M}}(\Phi)_J$.*

We now define the localized Rabin objective where we determine the winning frontiers based on the actual winning states in a BOMDP. We use the following auxiliary notation: We define the reachable frontier $RF := \mathsf{Reachable}(S) \cap F$. Then, we can determine where a local transition $s \xrightarrow{a} s'$ leads in the global system, $ToGlob_J(\langle s, a, s'\rangle) := \mathsf{BU}(\langle s, J\rangle, a, s')$ and finally consider $WinLocal_J(F, B) := \{f \in F \mid ToGlob_J(f) \in B\}$.

▶ **Definition 43** (Correct localized Rabin objective). *For belief J, the* correct localized Rabin objective *is $\Phi^{CLoc}(J) := \Phi^{Loc}(WinLocal_J(RF, Win_{\mathcal{M}}(\Phi)_{\subset J}))$.*

The notion of correctness in the definition above is justified by the following theorem, which says that computing the correct localized Rabin objective provides the belief-restricted winning region. That is, the winning region of the J-local MEMDP $\mathcal{M}\{J\}$ with its correct localized Rabin objective is equal to the global winning region restricted to J.

▶ **Theorem 44.** *For Rabin objective Φ: $(Win_{\mathcal{M}\{J\}}(\Phi^{CLoc}(J)) \cap S) \times \{J\} = Win_{\mathcal{M}}(\Phi)_J$.*

The theorem immediately leads to the following characterization of the winning region.

▶ **Corollary 45.** *For Rabin objective Φ: $Win_{\mathcal{M}}(\Phi) = \bigcup_J (Win_{\mathcal{M}\{J\}}(\Phi^{CLoc}(J)) \cap S) \times \{J\}$.*

Cor. 45 suggests computing the winning region from local MEMDPs. The computation can go bottom-up, as the winning region of a MEMDP restricted to a belief J only depends on the J-local MEMDP $\mathcal{M}\{J\}$ and the winning regions of beliefs $J' \subset J$. These observations lead us to Alg. 3. We construct the J-local MEMDP, recursively determine the winning status of all its frontier states, and then compute the local winning region of $\mathcal{M}\{J\}$.

▶ **Theorem 46.** *In Alg. 3 with Rabin objective Φ: $\textsc{Check}(\mathcal{M}, \Phi)$ iff $\iota \in Win_{\mathcal{M}}(\Phi)$.*

▶ **Lemma 47.** *Alg. 3 runs in polynomial space.*

This lemma follows from observing that a local MEMDP and thus its frontier is polynomial and that the recursion depth is limited by $|I|$. Thm. 46 and Lem. 47 together prove the main theorem Thm. 41: The decision problem of almost-sure Rabin objectives in MEMDPs is in PSPACE. Thus, almost-sure safety, Büchi, co-Büchi, and parity are in PSPACE too [13].

▶ **Theorem 48.** *The time complexity of Alg. 3 is in $O((|S|^2 \cdot |A|)^{|I|} \cdot poly(|\mathcal{M}|, |\Phi|))$.*

The bound in Thm. 48 is conservative[2], and it shows that deciding almost-sure Rabin objectives for 2-MEMDPs is in P. Almost-sure parity objectives for 2-MEMDPs were already known to be in P [36]. Indeed, it establishes the complexity for any fixed number of constants[3].

▶ **Corollary 49.** *For constant k, deciding almost-sure Rabin for k-MEMDPs is in* P.

7 Conclusion

We have presented a PSPACE algorithm for almost-sure Rabin objectives in MEMDPs. This result establishes PSPACE-completeness for many other almost-sure objectives, including parity, and completes the complexity landscape for MEMDPs. We additionally showed that all objectives under the possible semantics we consider in MEMDPs belong to the same complexity classes as MDPs. Interesting directions for future work are to investigate whether the constructions used in this paper can also be of benefit for quantitative objectives in MEMDPs or more expressive subclasses of POMDPs, for example, a form of MEMDPs where the environments may change over time.

References

1 Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *AAAI*, pages 2669–2678. AAAI Press, 2018.

2 Roman Andriushchenko, Milan Ceska, Sebastian Junges, Joost-Pieter Katoen, and Simon Stupinský. PAYNT: A tool for inductive synthesis of probabilistic programs. In *CAV (1)*, volume 12759 of *LNCS*, pages 856–869. Springer, 2021.

3 Sebastian Arming, Ezio Bartocci, Krishnendu Chatterjee, Joost-Pieter Katoen, and Ana Sokolova. Parameter-independent strategies for pMDPs via POMDPs. In *QEST*, volume 11024 of *LNCS*, pages 53–70. Springer, 2018.

4 Christel Baier, Nathalie Bertrand, and Marcus Größer. On decision problems for probabilistic Büchi automata. In *FoSSaCS*, volume 4962 of *LNCS*, pages 287–301. Springer, 2008.

5 Christel Baier, Marcus Größer, and Nathalie Bertrand. Probabilistic ω-automata. *J. ACM*, 59(1):1:1–1:52, 2012.

6 Christel Baier and Joost-Pieter Katoen. *Principles of model checking.* MIT Press, 2008.

7 Nathalie Bertrand, Patricia Bouyer, and Anirban Majumdar. Concurrent parameterized games. In *FSTTCS*, volume 150 of *LIPIcs*, pages 31:1–31:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.

8 Dietmar Berwanger and Laurent Doyen. On the power of imperfect information. In *FSTTCS*, volume 2 of *LIPIcs*, pages 73–82. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2008.

9 Iadine Chades, Josie Carwardine, Tara G. Martin, Samuel Nicol, Régis Sabbadin, and Olivier Buffet. MoMDPs: A solution for modelling adaptive management problems. In *AAAI*. AAAI Press, 2012.

10 Krishnendu Chatterjee, Martin Chmelík, Raghav Gupta, and Ayush Kanodia. Optimal cost almost-sure reachability in POMDPs. *Artif. Intell.*, 234:26–48, 2016.

11 Krishnendu Chatterjee, Martin Chmelík, Deep Karkhanis, Petr Novotný, and Amélie Royer. Multiple-environment Markov decision processes: Efficient analysis and applications. In *ICAPS*, pages 48–56. AAAI Press, 2020.

[2] A more precise bound can likely be obtained from the number of revealing transitions in the MEMDP.
[3] That is, the decidability problem is in XP with parameter *number of environments k*.

12 Krishnendu Chatterjee, Martin Chmelík, and Mathieu Tracol. What is decidable about partially observable Markov decision processes with ω-regular objectives. *J. Comput. Syst. Sci.*, 82(5):878–911, 2016.
13 Krishnendu Chatterjee, Luca de Alfaro, and Thomas A. Henzinger. The complexity of stochastic Rabin and Streett games'. In *ICALP*, volume 3580 of *LNCS*, pages 878–890. Springer, 2005.
14 Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Qualitative analysis of partially-observable Markov decision processes. In *MFCS*, volume 6281 of *LNCS*, pages 258–269. Springer, 2010.
15 Krishnendu Chatterjee, Marcin Jurdzinski, and Thomas A. Henzinger. Quantitative stochastic parity games. In *SODA*, pages 121–130. SIAM, 2004.
16 Krishnendu Chatterjee and Mathieu Tracol. Decidable problems for probabilistic automata on infinite words. In *LICS*, pages 185–194. IEEE Computer Society, 2012.
17 Min Chen, Emilio Frazzoli, David Hsu, and Wee Sun Lee. POMDP-lite for robust robot planning under uncertainty. In *ICRA*, pages 5427–5433. IEEE, 2016.
18 Murat Cubuktepe, Nils Jansen, Sebastian Junges, Joost-Pieter Katoen, and Ufuk Topcu. Convex optimization for parameter synthesis in MDPs. *IEEE Trans. Autom. Control.*, 67(12):6333–6348, 2022.
19 Luca de Alfaro. *Formal verification of probabilistic systems.* PhD thesis, Stanford University, USA, 1997.
20 Nathanaël Fijalkow. What is known about the value 1 problem for probabilistic automata? *CoRR*, abs/1410.3770, 2014.
21 Nathanaël Fijalkow, Nathalie Bertrand, Patricia Bouyer-Decitre, Romain Brenguier, Arnaud Carayol, John Fearnley, Hugo Gimbert, Florian Horn, Rasmus Ibsen-Jensen, Nicolas Markey, Benjamin Monmege, Petr Novotný, Mickael Randour, Ocan Sankur, Sylvain Schmitz, Olivier Serre, and Mateusz Skomra. Games on graphs. *CoRR*, abs/2305.10546, 2023.
22 Nathanaël Fijalkow, Hugo Gimbert, Edon Kelmendi, and Youssouf Oualhadj. Deciding the value 1 problem for probabilistic leaktight automata. *Log. Methods Comput. Sci.*, 11(2), 2015.
23 Nathanaël Fijalkow, Hugo Gimbert, and Youssouf Oualhadj. Deciding the value 1 problem for probabilistic leaktight automata. In *LICS*, pages 295–304. IEEE Computer Society, 2012.
24 Vojtech Forejt, Marta Z. Kwiatkowska, Gethin Norman, and David Parker. Automated verification techniques for probabilistic systems. In *SFM*, volume 6659 of *LNCS*, pages 53–113. Springer, 2011.
25 Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.*, 16:1437–1480, 2015.
26 Hugo Gimbert and Youssouf Oualhadj. Probabilistic automata on finite words: Decidable and undecidable problems. In *ICALP (2)*, volume 6199 of *LNCS*, pages 527–538. Springer, 2010.
27 Garud N. Iyengar. Robust dynamic programming. *Math. Oper. Res.*, 30(2):257–280, 2005.
28 Nils Jansen, Sebastian Junges, and Joost-Pieter Katoen. Parameter synthesis in Markov models: A gentle survey. In *Principles of Systems Design*, volume 13660 of *LNCS*, pages 407–437. Springer, 2022.
29 Sebastian Junges, Nils Jansen, and Sanjit A. Seshia. Enforcing almost-sure reachability in POMDPs. In *CAV (2)*, volume 12760 of *LNCS*, pages 602–625. Springer, 2021.
30 Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101(1-2):99–134, 1998.
31 Nicolas Meuleau, Kee-Eung Kim, Leslie Pack Kaelbling, and Anthony R. Cassandra. Solving POMDPs by searching the space of finite policies. In *UAI*, pages 417–426. Morgan Kaufmann, 1999.
32 Arnab Nilim and Laurent El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Oper. Res.*, 53(5):780–798, 2005.
33 Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of Markov decision processes. *Math. Oper. Res.*, 12(3):441–450, 1987.

34 Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994.

35 Jean-François Raskin, Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Algorithms for omega-regular games with imperfect information. *Log. Methods Comput. Sci.*, 3(3), 2007.

36 Jean-François Raskin and Ocan Sankur. Multiple-environment Markov decision processes. In *FSTTCS*, volume 29 of *LIPIcs*, pages 531–543. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2014.

37 Marc Rigter, Bruno Lacerda, and Nick Hawes. Minimax regret optimisation for robust planning in uncertain Markov decision processes. In *AAAI*, pages 11930–11938. AAAI Press, 2021.

38 Marnix Suilen, Marck van der Vegt, and Sebastian Junges. A PSPACE algorithm for almost-sure Rabin objectives in multi-environment MDPs, 2024. `arXiv:2407.07006`.

39 Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998.

40 Marck van der Vegt, Nils Jansen, and Sebastian Junges. Robust almost-sure reachability in multi-environment MDPs. In *TACAS (1)*, volume 13993 of *LNCS*, pages 508–526. Springer, 2023.

41 Tobias Winkler, Sebastian Junges, Guillermo A. Pérez, and Joost-Pieter Katoen. On the complexity of reachability in parametric Markov decision processes. In *CONCUR*, volume 140 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.