# Reduction Techniques for Model Checking and Learning in MDPs

**Suda Bharadwaj*** and **Stéphane Le Roux†** and **Guillermo A. Pérez†** and **Ufuk Topcu***
*University of Texas at Austin
†Université libre de Bruxelles
{suda.b, utopcu}@utexas.edu, {stephane.le.roux, gperezme}@ulb.ac.be

## Abstract

Omega-regular objectives in Markov decision processes (MDPs) reduce to reachability: find a policy which maximizes the probability of reaching a target set of states. Given an MDP, an initial distribution, and a target set of states, such a policy can be computed by most probabilistic model checking tools. If the MDP is only partially specified, i.e., some probabilities are unknown, then model-learning techniques can be used to statistically approximate the probabilities and enable the computation of the desired policy. For fully specified MDPs, reducing the size of the MDP translates into faster model checking; for partially specified MDPs, into faster learning. We provide reduction techniques that allow us to remove irrelevant transition probabilities: transition probabilities (known, or to be learned) that do not influence the maximal reachability probability. Among other applications, these reductions can be seen as a pre-processing of MDPs before model checking or as a way to reduce the number of experiments required to obtain a good approximation of an unknown MDP.

## 1 Introduction

Markov decision processes (MDPs) are one of the most widely used tools for modelling decision-making under uncertainty [Littman, 1996; Papadimitriou and Tsitsiklis, 1987]. They are widely used in areas of planning [Russell and Norvig, 2010; Ding *et al.*, 2014], model-based reinforcement learning [Strehl *et al.*, 2009], formal verification [Baier and Katoen, 2008], robotics [Lahijanian *et al.*, 2010], and control [Bernstein *et al.*, 2002]. Arguably the most fundamental question one can ask in an MDP is to maximize the probability of reaching some target set of states from a set of initial states. That is, to compute the maximal reachability probability and a policy which — when implemented on the MDP — achieves that probability. In the artificial intelligence planning community, this problem is also known as MAXPROB (see, *e.g.*, [Kolobov *et al.*, 2011; Steinmetz *et al.*, 2016]). A sample instance of the associated decision problem is as follows: "is there a policy to ensure the probability of reaching state $t$ from state $s$ is at least $10\%$?" More elaborate objectives defined with respect to reward or cost functions have also been considered in the literature [Puterman, 2005]. It can, however, be cumbersome to map complex goals to an appropriate reward structure in an MDP and thus find a control policy. To this end, linear temporal logic (LTL) has been commonly used as a way to formally specify high-level goals (see, *e.g.*, [Svorenová *et al.*, 2013; Lacerda *et al.*, 2015]). It is known that maximizing the probability of satisfying several objectives, notably LTL objectives, reduces to a reachability objective [Baier and Katoen, 2008].

In this paper we develop MDP reduction techniques that preserve the maximal reachability probability. We first discuss why such reductions are needed.

**Need for reductions in model checking.** Model checking is a technique to formally verify properties of a model of a system against a formal specification, given, for instance, as an LTL formula [Courcoubetis and Yannakakis, 1995]. The advantage of model checking is that it provides an automated way for the user to verify if their specification holds in a model. Given an MDP, initial states, and a target set of states, model checking tools are able to compute the maximal reachability probability in time polynomial in the size of the MDP [Kwiatkowska *et al.*, 2011; Dehnert *et al.*, 2017].

Unfortunately, model checking is a model-based technique: it assumes the MDP will be given in a completely specified manner, *i.e.* all transition probabilities are assumed to be known in advance. In practice, these values often come from simulations or they are estimated by hand. If the accuracy of these values is unknown, then the result of model checking carries no relevant insight. Even if the transition probabilities have some guaranteed probability of being accurate, the guarantees provided by model checking will depend on both the precision and confidence parameters: the output will be only probably approximately correct (PAC) [Valiant, 2013].

One way to mitigate these weaknesses is to determine which transition probabilities affect the maximal reachability probability. It is known that not all transition probabilities impact the maximal reachability probability [Ciesinski *et al.*, 2008; Brázdil *et al.*, 2014]. If some transition probabilities are *irrelevant*, then the guarantees on the outcome of the model checking process do not depend on the precision or confidence related to those values.

**Need for reductions in learning.** Another way to deal with partially specified MDPs is to use learning techniques [Kael-

bling *et al.*, 1996]. In this approach, and in order to obtain a PAC output, the model (*i.e.*, its transition probabilities) are learned up to some precision with some desired confidence and then the MDP is processed [Wen and Topcu, 2016; Fu and Topcu, 2014]. A drawback of learning is that it requires a large number of samples — and, in turn, a large amount a time — to learn the transition probabilities with desirable precision and confidence parameters [Kawaguchi, 2016; Kolter and Ng, 2009; Russell *et al.*, 2015]. This is especially the case for PAC learning algorithms like R-max [Brafman and Tennenholtz, 2003] and $E^3$ [Kearns and Singh, 2002]. While these techniques provide attractive theoretical guarantees on optimality, they require practically excessive amounts of sampling and exploration [Kawaguchi, 2016; Kolter and Ng, 2009; Russell *et al.*, 2015].

Here, it is natural to ask which transition probabilities one really has to learn and which ones can be left unexplored. If some transition probabilities are known to have no effect on the maximal reachability probability, then they can be omitted from the learning phase without affecting the guarantees on the output. We show that if the objective can be reduced to reachability, then the entire MDP need not be sampled and the MDP can be reduced to shorten necessary learning time and still provide the same guarantees.

**Our contribution.** We provide reduction techniques for MDPs to allow for more efficient reachability analysis and learning without sacrificing any correctness. In other words, given an MDP, initial states, and target states, we compute — in polynomial time — a smaller MDP with equivalent maximal reachability probability. We do this by removing distributions — state-action pairs and the resulting transitions enabled with non-zero probability — in the MDP that do not affect the maximal probability of reaching a target set of states from some initial states. More precisely, our algorithm uses the graph structure of the MDP (and can thus be used even if all the probabilities are unknown) to determine if some distributions are 'better' than others. We then use this information to remove sub-optimal actions (and, hence, distributions) from the MDP. We illustrate the reductions using small examples and show that we achieve reductions on MDPs with no end components and unique extremal probability states (these notions will be made precise in the sequel).

In two case studies from the well-known probabilistic model checking tool PRISM, the reduction in the size of the MDPs obtained by applying our techniques far outperforms current reduction techniques. We also test the proposed techniques on a model-based learning application with a classical gridworld test setup and observe a large reduction in learning time.

**Related work.** Reductions in MDPs have been investigated in formal verification. In [Ciesinski *et al.*, 2008], the authors collapse end components and states from which the probability of satisfying the specification is 0 or 1. These will be presented in Sec. 3. The same techniques were also mentioned in [Brázdil *et al.*, 2014], where learning techniques are introduced to aid model checking MDPs. The authors also provide an algorithm to compute end components on the fly in the case where they are not known beforehand. We assume, however, that the graph structure of the MDP is known beforehand and hence all end components can be computed.

**Structure.** In Sec. 2, we provide the definitions and notation that will be followed in the paper. We present the known reduction techniques from related works in Sec. 3, followed by our novel techniques in Sec. 4. In Sec. 5, we present experiments where the reductions were applied on established model checking case studies from the PRISM tool as well as learning in a gridworld. We conclude and provide future directions in Sec. 6.

## 2 Preliminaries

We follow notation from [Littman, 1996; Puterman, 2005].

### 2.1 Markov Decision Processes

We denote by $\mathcal{D}(X)$ the set of all probability distributions on a finite set $X$, *i.e.* all functions $f : X \to [0, 1]$ such that $\sum_{x \in X} f(x) = 1$. For $f \in \mathcal{D}(X)$ we denote by $\mathsf{supp}(f)$ the *support* of $f$. That is, the set $\{x \in X \mid f(x) > 0\}$.

**Definition 1.** *An* MDP *is a tuple* $M = (S, \mathcal{A}, \delta)$ *where* $S$ *is a finite set of* states*,* $\mathcal{A}$ *is a finite alphabet of* actions*,* $\delta : S \times \mathcal{A} \to \mathcal{D}(S)$ *is a (partial)* probabilistic transition function *that assigns to a state* $s$ *and an action* $a \in \mathcal{A}$ *a probability distribution over the successor states. We abbreviate* $\delta(s, a)(s')$ *by* $\delta(s'|s, a)$.

Pictorially, the states of an MDP will be represented by circles; a distribution $\delta(s, a)$, by an arrow leaving state $s$ with multiple heads — one per state $s' \in \mathsf{supp}(\delta(s, a))$ — having its tail labelled with $a$ and every head pointing to $s'$ with the value $\delta(s'|s, a)$.

**Same-support functions.** Two probabilistic transition functions $\delta, \delta'$ are said to have the same support if $\mathsf{supp}(\delta(s, a)) = \mathsf{supp}(\delta'(s, a))$ holds for all pairs $(s, a) \in S \times \mathcal{A}$. We write $\mathsf{supp}(\delta) = \mathsf{supp}(\delta')$ to denote this.

**Runs and policies.** A *run* from state $s_0$ is a (possibly infinite) sequence $\varrho = s_0 a_0 s_1 a_1 s_2 a_2 \ldots$ of states and actions such that for all $i \geq 0$ we have $\delta(s_{i+1}|s_i, a_i) > 0$. A *policy* corresponds to a way of selecting actions based on the history of states and actions. We focus on *deterministic stationary* policies, since these are known to be sufficient for reachability probability optimization [Condon, 1992; Puterman, 2005]. Formally, a (deterministic stationary) policy is a function $\pi : S \to \mathcal{A}$ which assigns to every state an action.

A run $\varrho$ is *consistent* with a policy $\pi$ if it can be obtained by extending its finite prefixes using $\pi$. Formally, $\varrho = s_0 a_0 s_1 a_1 \ldots$ is consistent with $\pi$ if for all $i \geq 0$ we have that $a_i = \pi(s_i)$ and $\delta(s_{i+1}|s_i, a_i) > 0$.

**Reachability probability.** Given an initial distribution $\iota \in \mathcal{D}(S)$, a policy $\pi$, and a target set of states $T \subseteq S$, the *reachability probability* $\mathcal{P}_\iota^{\delta,\pi}[\mathrm{Reach}(T)]$ of $\pi$ is the probability that a run starting from a state $s$, sampled from $\iota$, and consistent with $\pi$ will reach a state from $T$. This definition can be formalized by a standard construction of a probability measure induced by $\pi$ over the set of all runs (see, *e.g.*, [Puterman, 2005]). The *maximal reachability probability* is $\mathrm{Val}_\iota^\delta(T) := \max_\pi \mathcal{P}_\iota^{\delta,\pi}[\mathrm{Reach}(T)]$.

**Irrelevant distributions.** For every initial distribution $\iota$ and target set $T$, we are interested in the set of state-action pairs

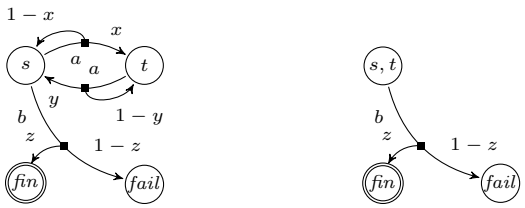


Figure 1: Two MDPs with action alphabet $\mathcal{A} = \{a, b\}$: an MDP with EC $(\{s, t\}, \{s \mapsto a, t \mapsto a\})$, on the left, and the same MDP after collapsing ECs, on the right

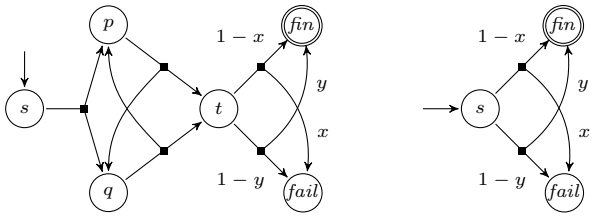


Figure 2: Two MDPs shown, for clarity, actions and some probabilities have been omitted: an MDP with no ECs (on the left), and a smaller MDP with equivalent maximal reachability probability from $s$ to *fin* (on the right)

$(s, a) \in S \times \mathcal{A}$ for which changing the distribution $\delta(s, a)$ — without changing its support — will not change the value $\mathrm{Val}^{\delta}_{\iota}(T)$. Formally, we consider an arbitrary probabilistic transition function $\delta'$ such that: $(i)$ $\delta(t, b) = \delta'(t, b)$ for all $(t, b) \neq (s, a)$ and $(ii)$ $\mathsf{supp}(\delta(s, a)) = \mathsf{supp}(\delta'(s, a))$. We say $\delta(s, a)$ is *irrelevant* if, for all such $\delta'$, it holds that $\mathrm{Val}^{\delta'}_{\iota}(T) = \mathrm{Val}^{\delta}_{\iota}(T)$ — where $\mathrm{Val}^{\delta'}_{\iota}(T)$ refers to the maximal reachability probability in the MDP $(S, \mathcal{A}, \delta')$.

## 3 Well-Known Reductions

We recall the most widely used reductions applied to MDPs before computing maximal reachability probability from an initial state to a target set of states. Note that there are other obvious optimizations, such as removal of states not reachable from the initial state, that we do not describe here.

For the remainder of this section, we consider a given MDP $M = (S, \mathcal{A}, \delta)$ and a target set of states $T$.

### 3.1 End Components

An *end component* (EC) is a pair $(Q, \alpha)$ where $Q \subseteq S$ and $\alpha : Q \to 2^{\mathcal{A}}$ is a mapping from states to actions such that: by playing an action $\alpha(q)$ from state $q \in Q$, with probability 1, the next state reached will also be in $Q$. Formally, we require that for all $q \in Q$, the following holds: $(i)$ $\alpha(q) \subseteq \mathcal{A}$ is non-empty. $(ii)$ If there are $q' \in S$ and $a \in \alpha(q)$ such that $\delta(q'|q, a) > 0$ then $q' \in Q$. $(iii)$ For all $q' \in Q$ there is a run $s_0 a_0 \dots a_{n-1} s_n$ from $q$ going to $q'$ (*i.e.*, $s_0 = q$ and $s_n = q'$) such that $a_i \in \alpha(s_i)$ for all $0 \leq i < n$.

**Lemma 1** (From [De Alfaro, 1997])**.** *In an EC of an MDP, for all states, there are policies to reach any other state in the EC with probability* 1.

**Collapsing ECs.** The collapsing of ECs into a single state is a common optimization used in tools for model checking LTL properties in MDPs [Baier and Katoen, 2008; Ciesinski *et al.*, 2008]. Intuitively, this can be done because all the states from the same EC have the same maximal reachability probability (this follows from Lemma 1).

**Proposition 1.** *For all ECs $(Q, \alpha)$, for all $q \in Q$, and for all $a \in \alpha(q)$, the distribution $\delta(q, a)$ is irrelevant.*

See Fig. 1 for an example of how collapsing EC removes irrelevant distributions.

### 3.2 Extremal Probability States

Transitions from states with maximal reachability probability 0 or 1 are also irrelevant. Both sets of states can be computed, even if the probabilities are not known, using graph-based algorithms [Courcoubetis and Yannakakis, 1995]. This optimization is also well-known and standard in probabilistic model checking tools. Since, for such states, either a policy exists such that we certainly reach the target set or there is no way to reach the target set of states, clearly transitions leaving them (more precisely, the transitions defined by state-action pairs with them) are irrelevant.

**Proposition 2.** *For all $(s, a) \in S \times \mathcal{A}$ such that $\mathrm{Val}^{\delta}_{\{s \mapsto 1\}}(T) \in \{0, 1\}$, the distribution $\delta(s, a)$ is irrelevant.*

**Collapsing extremal probability states.** The above result allows us to assume, without loss of generality, that the MDP has exactly one unique target state *fin* with maximal reachability probability 1. (If this is not the case, then we redirect all transitions going to those states so that they lead to *fin*.) Similarly, we can assume there is exactly one state, *fail*, with maximal reachability probability 0.

**Example 1** (Other irrelevant distributions)**.** *Consider the left-hand MDP shown in Fig. 2. Although the transition probabilities for transitions among states $s, p, q, t$ are not given, it is clear that for any policy $\pi$, we have $\mathcal{P}^{\delta,\pi}_{\{s \mapsto 1\}}[\mathrm{Reach}(\{t\})] = 1$. In words, $t$ is unavoidable from $s$, regardless of the probability values that are not shown. Hence, the maximal reachability probability — with respect to $s$ as initial state and fin as target state — only depends on the choice of action from $t$. Thus, only distributions $\delta(t, a)$, for $a \in \mathcal{A}$, are not irrelevant, and the MDP can be simplified by transforming it into the right-hand MDP in the figure. Unfortunately the states do not form an EC and, therefore, we cannot obtain the right-hand MDP using the existing reduction rules only.*

In the next section we will describe new reduction rules which help, in particular, to achieve the transformation described in the previous example.

## 4 New Reductions

We now describe new reductions based on a binary relation on the distributions that can be inferred from the directed graph of a given MDP $M = (S, \mathcal{A}, \delta)$ and target state *fin*.

### 4.1 A Preorder on Distributions

For any two state-action pairs $(p, a), (q, b) \in S \times \mathcal{A}$, we say $(q, b)$ is *always (strictly) better* than $(p, a)$ if, for all probabilistic transition functions $\delta'$ with $\mathsf{supp}(\delta') = \mathsf{supp}(\delta)$,

$$\mathrm{Val}^{\delta'}_{\delta'(p,a)}(\mathit{fin}) \leq \mathrm{Val}^{\delta'}_{\delta'(q,b)}(\mathit{fin}) \qquad (1)$$

(resp. $<$). We then write $(p,a) \trianglelefteq (q,b)$ (resp. $\triangleleft$) to denote the fact. Given a state-action pair $(p,a)$ and a set $P \subseteq S \times \mathcal{A}$, we lift the relation $\trianglelefteq$ (resp. $\triangleleft$) to sets by letting $(p,a) \preceq P$ (resp. $\prec$) denote the fact that, for all probabilistic transition functions $\delta'$ with the same support as $\delta$, there exists some $(q,b) \in P$ such that Equation (1) holds.

**Lemma 2.** *The binary relation $\trianglelefteq$ is a preorder. If $(p,a) \prec P$ (resp. $\preceq$) and $P \subseteq P'$, then $(p,a) \prec P'$ (resp. $\preceq$).*

**Choosing actions based on the preorder.** The main idea behind how we will use the relations $\triangleleft, \prec$ is quite intuitive. If, at a state $s$, there is a choice of playing actions $a, b \in \mathcal{A}$ and $(s,b)$ is always better than $(s,a)$, then we remove the possibility of choosing action $a$. More precisely, if $(s,a) \prec \{(s,b) \mid b \in \mathcal{A}\}$, then we can set $\delta(s'|s,a)$ to 0 for all $s'$. Regardless of the actual values of the probabilities, any policy maximizing the reachability probability will not play $a$.

**Probability-1 shortcuts.** For convenience, in the sequel we assume that for all states $s, t \in S$, if $\mathrm{Val}^{\delta}_{\{s \mapsto 1\}}(\{t\}) = 1$, then for all $(t',a) \in S \times \mathcal{A}$ with $\delta(t'|t,a) > 0$ there is some $b \in \mathcal{A}$ such that $\delta(t'|s,b) = \delta(t'|t,a)$. This assumption is at no loss of generality. If it does not hold, we can apply (in polynomial time) the following pre-processing to the MDP. If from $s$ there is some policy to ensure reaching $t$ with probability 1, then using fresh actions we add shortcuts from $s$ to the distributions $\delta(t,a)$ that can be 'chosen' from $t$.

**Example 2.** *In the left-hand MDP from Fig. 2, we would add shortcuts from $s$ to both distributions available from $t$. Note that to obtain the desired transformation for this MDP, all that remains is to somehow establish that the newly added shortcuts are always better than the unique state-action pair that was originally available from $s$ (to justify removing the latter).*

In what follows we give two sufficient conditions for establishing the 'always better than' relation.

## 4.2 Sufficient Conditions for the Preorder

Both conditions rely on the notion of *separating set*.

**Separating sets.** Consider a set of state-action pairs $P \subseteq S \times \mathcal{A}$. If all runs starting from a state sampled from $\iota$ and reaching the state $t \in S$ go through some state-action pair from $P$, we say that $P$ *separates* $t$ from $\iota$.

**Separating sets for *fin*.** Suppose we are given a state-action pair $(s,a) \in S \times \mathcal{A}$ and a set $M \subseteq S \times \mathcal{A}$, and we are asked whether $(s,a) \preceq M$ holds. Denote by $P^{\prec}(M)$ the set of state-action pairs for which there exists some always-better pair in $M$, *i.e.* the set $\{(t,b) \in S \times \mathcal{A}\} \mid (t,b) \prec M\}$. We now claim that if $M \cup P^{\prec}(M)$ separates *fin* from $\delta(s,a)$, then $(s,a) \preceq M$ holds.

**Theorem 1.** *If $M \cup P^{\prec}(M)$ separates fin from $\delta(s,a)$, then $(s,a) \preceq M$. If additionally, $\mathrm{Val}^{\delta}_{\delta(s,a)}(\{t \mid \exists b \in \mathcal{A} : (t,b) \in M\}) < 1$, then $(s,a) \prec M$.*

Since $M \cup P^{\prec}(M)$ separates *fin* from $\delta(s,a)$, then for all $\delta'$ with the same support as $\delta$ we have that $\mathrm{Val}^{\delta'}_{\delta'(s,a)}(\{fin\})$ must be equal to a convex combination of the values $\{\mathrm{Val}^{\delta'}_{\delta'(t,b)}(\{fin\}) \mid (t,b) \in M \cup P^{\prec}(M)\}$. Indeed, by definition of separating sets, any run reaching *fin* from a state in the support of $\delta'(s,a)$ will first reach a state from $\{t \mid (t,b) \in (M \cup P^{\prec}(M))\}$. It is then easy to verify that there exists $(t,b) \in (M \cup P^{\prec}(M))$ such that $\mathrm{Val}^{\delta'}_{\delta'(s,a)}(\{fin\}) \leq \mathrm{Val}^{\delta'}_{\delta'(t,b)}(\{fin\})$, which gives us the first part of the claim. If we also know there is no policy to reach $T := \{t \mid (t,b) \in M\}$ from $\delta(s,a)$ with probability 1, we can then use the definition of $P^{\prec}(M)$ to show that the inequality is strict.

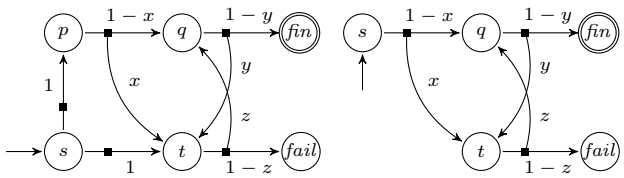


Figure 3: Two MDPs with the same maximal reachability probability from $s$ to *fin* (for clarity, actions not shown); the MDP on the right is obtained via Thm. 2

**Example 3.** *Consider again the left-hand MDP from Fig. 2. Recall that we have added shortcuts from $s$ to both distributions available from $t$ (see Ex. 2). Let us denote those two distributions by $\delta(t,a)$ and $\delta(t,b)$. We can now use Thm. 1 to justify removing the only transition available from $s$ originally. Let us denote the latter by $\delta(s,a)$. Clearly, if we remove $t$ from the MDP, then there are no runs which can reach fin from $\delta(s,a)$. Hence, $\{(t,a),(t,b)\}$ separates $s$ from fin. It follows that either $(t,a)$ or $(t,b)$ is always better than $(s,a)$, and that we can remove $\delta(s,a)$.*

**Separating sets for *fail*.** Suppose we are given two state-action pairs $(s,a),(t,b) \in S \times \mathcal{A}$ and we are asked whether $(t,b) \trianglelefteq (s,a)$ holds. Let $P^{\trianglerighteq}(t,b)$ denote the set $\{(u,c) \in S \times \mathcal{A} \mid (t,b) \triangleleft (u,c)\}$, and $P^{\triangleright}(t,b)$ be defined similarly. We now claim that if there is a policy to reach $P^{\trianglerighteq}(t,b) \cup \{t, fin\}$ from $\delta(s,a)$ with probability 1, then $(t,b) \trianglelefteq (s,a)$ holds.

**Theorem 2.**
- *If* $\mathrm{Val}^{\delta}_{\delta(s,a)}(\{u \mid (u,c) \in P^{\trianglerighteq}(t,b)\} \cup \{t, fin\}) = 1$, *then* $(t,b) \trianglelefteq (s,a)$.
- *If* $\mathrm{Val}^{\delta}_{\delta(s,a)}(\{u \mid (u,c) \in P^{\triangleright}(t,b)\} \cup \{fin\}) = 1$ *also holds, then* $(t,b) \triangleleft (s,a)$.

The intuition why the result holds is as follows. Since there is a policy to ensure reaching $\{u \mid (u,c) \in P^{\trianglerighteq}(t,b)\} \cup \{t, fin\}$ with probability 1, the maximal reachability probability from $\delta(s,a)$ must be greater than the least maximal reachability probability from a state in that set. By definition of $P^{\trianglerighteq}(t,b)$, all its elements are always better than $(t,b)$. It follows that $(s,a)$ is also always better than $(t,b)$. An alternative way to see it is to notice that $\{u \mid (u,c) \in P^{\trianglerighteq}(t,b)\} \cup \{t, fin\}$ in fact separates *fail* from $\delta(s,a)$ — under some policy. Hence, all runs from $\delta(s,a)$ which reach *fail* pass through the set (which should therefore be avoided if possible).

**Example 4.** *Consider the left-hand MDP from Fig. 3. Let us denote by $\delta(s,a)$ the distribution assigning probability 1 to state $p$; by $\delta(s,b)$ the one assigning probability 1 to $t$; by $\delta(p,a)$ and $\delta(t,a)$ the ones with probability $1-x$ and $1-z$*

*of reaching $q$ and $fail$, respectively. Observe that we have probability-1 shortcuts from $s$ to $\delta(p,a)$ and to $\delta(t,a)$. All that is left, to obtain the reduced MDP on the right side of the figure, is to argue that $(p,a)$ is always better than the other three pairs so that they can be removed. To do so, we note that from all of them, the only way to reach $fail$ is to go through $\delta(t,a)$. The desired relation then holds by Thm. 2.*

**Algorithm 1** MDP reductions based on irrelevant distributions

```
1: procedure REDUCE
2:     Collapse maximal end components
3:     Merge prob. 1 and 0 states
4:     Create probability-1 shortcuts
5:     alBetter' ← ∅
6:     repeat
7:         alBetter ← alBetter'
8:         for (s,a) ∈ S × A do
9:             M ← {(s,b) | b ≠ a}
10:            if isAlwaysBetter(M,(s,a)) then
11:                alBetter' ← alBetter' ∪ {(M,(s,a))}
12:                remove (s,a)
13:                break      // to avoid removing all actions
14:            end if
15:        end for
16:    until alBetter' = alBetter
17: end procedure
```

### 4.3 Summary of the Algorithm

Our proposed reduction algorithm (see Alg. 1) consists in first applying the well-known reductions. Second, create probability-1 shortcuts. Third, for each state $s$ we check whether for some action $a$, $(s,a)$ is always worse than all alternatives $(s,b)$; if this is the case, then we remove $\delta(s,a)$. To determine whether this is the case, we use one of the two conditions described above. More precisely, the *isAlwaysBetter* function checks the conditions from Thms. 1 and 2 based on the computed relation *alBetter* — which is initially empty.

Note that the third step is repeated until convergence. The number of iterations before convergence will be linear in the size of the MDP since at least one transition is removed after every iteration.

Recall that after these reductions are applied, the maximal reachability probability in the reduced MDP is equivalent to the maximal reachability probability in the original MDP. Furthermore, both conditions given by Thms. 1 and 2 can be checked in polynomial time. This is because they rely only on deciding whether a given set is a separating set and deciding whether the maximal reachability probability of a distribution is not 0 or 1. Since we repeat the process until convergence, *i.e.* at most a linear number of times, and the checks in each iteration are computable in polynomial time, the entire procedure takes polynomial time.

## 5 Experiments

We present some results of running the reduction techniques on two model checking case studies from PRISM [Kwiatkowska *et al.*, 2011], a well-known probabilistic model checking tool, as well as a model-learning task in a gridworld.

| Formula | No reds. | Known reds. | New reds. |
|---|---|---|---|
| $\varphi_1$ | 400 | 392 | 76 |
| $\varphi_2$ | 400 | 392 | 92 |

Table 1: Size of the MDP after applying no reductions, currently known reductions, and the proposed reductions. The numbers correspond to the number of distributions left in the reduced MDP.

| Params. | No reds. | Known reds. | New reds. |
|---|---|---|---|
| K = 1 | 553 | 530 | 59 |
| K = 2 | 827 | 804 | 105 |

Table 2: Size of the Zeroconf protocol MDP after applying no reductions, the currently known reductions, and our proposed reductions.

### 5.1 Model Checking Experiments

Currently, the most widely used algorithm to model check MDPs in practice is value iteration (see , e.g., [Kwiatkowska *et al.*, 2011; Haddad and Monmege, 2014]). The known upper bound on the number of iterations required for the value obtained to satisfy any formal guarantees is polynomial in the size of the representation of the smallest transition probability of the MDP. This makes reporting running times for value iteration before and after reductions not interesting: in any MDP with irrelevant distributions, we can make the probabilities on those distributions as small as needed to make value iteration (before applying our reductions) take as long as desired.

**Randomized Consensus Shared Coin Protocol.** We look at a model of an asynchronous shared coin protocol detailed in [Aspnes and Herlihy, 1990]. The goal is to compute the maximal probability of the following two LTL formulas: $\varphi_1 = \Diamond$ ("finished" $\wedge$ "all coins equal 1") and $\varphi_2 = \Diamond$ ("finished" $\wedge \neg$"all coins equal 1") — where $\Diamond$ should be read as *eventually*. The protocol is modelled in the PRISM language by [Kwiatkowska *et al.*, 2001].[1] We implement our reduction techniques, as well as the currently-used ones, and the results are given in Tab. 1.

**IPv4 Zeroconf Protocol.** This problem involves checking for the probability of the host choosing an IP address that is already in use. The corresponding formula is $\Diamond(l = 4 \wedge ip = 1)$. The protocol is modelled by [Kwiatkowska *et al.*, 2004].[1] Results of applying the reductions are shown in Tab. 2.

**Summary of results.** The algorithm results in an $81\%$ and $77\%$ reduction in the size of the randomized consensus MDP compared to the $2\%$ under existing techniques. For the Zeroconf protocol MDP, we were able to achieve $89.3\%$ reduction compared to $4.2\%$ under existing techniques. For both cases, the corresponding LTL formula was verified in the resulting fully reduced MDP. For the randomized consensus and Zeroconf MDPs the maximal probability of satisfying the respective LTL formulas on the reduced MDPs was found to

[1] Model details can be found on the PRISM case studies website. We use model parameters: 2 coins and $K = 2$, and $reset = \text{TRUE}$ and $K \in \{1, 2\}$, respectively.

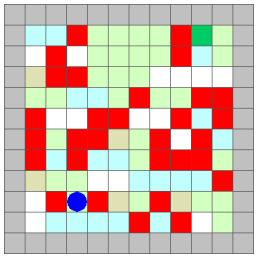

Figure 4: A sample gridworld case study for learning. The different colours of the cells correspond to different terrains which affect the transition probabilities.

| No. states | No reds. | Known reds. | New reds. |
|---|---|---|---|
| 144 | 564 | 273 | 166 |
| 225 | 900 | 423 | 288 |
| 400 | 1600 | 738 | 570 |

Table 3: Average reduction results for three gridworlds: a $12 \times 12$, a $15 \times 15$, and a $20 \times 20$

be $0.55$, $0.062$ and $2.9 \times 10^{-4}, 5.3 \times 10^{-5}$. This matches the values obtained from PRISM.

### 5.2 Reductions in Gridworlds with LTL Objective

We present an example of a learning problem in a gridworld. For the robot's different actions (heading north ('N'), south ('S'), west ('W') and east ('E')), the probability of arriving at the correct cell is in a certain range: $[0.75, 0.80]$ in this example. With a relatively small probability, the robot will arrive at the cell adjacent to the intended one. For example, with action 'N', the intended cell is the one to the north ('N'), whose the adjacent ones are the northeast ('NE') and northwest cells ('NW'). We look at the specific example shown in Fig. 4.

**LTL objective.** The objective is to maximize the probability of reaching the green state while avoiding the red ones: $\Diamond R_1 \wedge \Box \neg R_2$ where $R_1$ is the green state and $R_2$ is the set of all the red states in the gridworld. The values of the transition probabilities are not known. The underlying task is to learn these probabilities and compute a policy that maximizes the probability of reaching the target. We use a PAC-MDP learning algorithm similar to that shown in [Fu and Topcu, 2014]. In order to mitigate the high sampling requirement and learning time mentioned earlier, we apply the proposed reduction technique to reduce the distributions that need to be sampled without sacrificing the PAC guarantees.

**Experiment setup.** We test three differently sized gridworlds. For each gridworld we randomly distribute a given number of obstacles, and then apply the reduction techniques. For standardization, $20\%$ of each gridworld was populated with obstacles. Tab. 3 summarizes the average reduction we observe after several runs with a randomized distribution of obstacles in the grid for each run.

**Average reduction amount.** On average, the size of the fully reduced MDP is about $70\%$ smaller than the original MDP in all three gridworld sizes. The currently applied techniques reduced the size by about $50\%$ in all cases.

| | No reds. | Known reds. | New reds. |
|---|---|---|---|
| Distributions | 400 | 102 | 8 |
| Episodes | 1,133,243 | 948,882 | 83,564 |
| Total steps | 11,683,438 | 7,848,560 | 734,465 |

Table 4: Number of steps and learning episodes needed by the robot, in one $10 \times 10$ gridworld, until $\varepsilon$-optimality can be guaranteed in three cases: original MDP, after known techniques are applied, and after our techniques are applied.

### 5.3 Learning in Gridworlds with LTL Objectives

We study the impact of reductions on the learning process.

**PAC learning.** We now run a modified version of the R-max learning algorithm presented in [Fu and Topcu, 2014] on one of the reduced $10 \times 10$ MDPs. Explicitly, we aim to learn a policy that with probability at least $1 - \delta$ will be $\varepsilon$-optimal in maximizing reachability probability. We measure the decrease in sampling required before $\varepsilon$-optimality is achieved for $\varepsilon = 0.01$ and $\delta = 0.05$.

**Learning results.** The number of steps needed for the learning phase is reduced by $94\%$ when the full reductions are implemented compared to $67\%$ under existing techniques.

## 6 Conclusion and Future Work

We have proposed a notion of irrelevant distribution to formalize the set of transition probabilities that do not affect the maximal reachability probability. Further, we argue that known MDP reductions, such as collapsing ECs, are in fact removing some irrelevant distributions. We have also described a new algorithm to remove irrelevant distributions which had not, to the best of our knowledge, been considered before. Determining whether our algorithm guarantees that no irrelevant distributions are left, is an open problem.

Our algorithm has been implemented in a prototype tool and empirical results suggest that our algorithm achieves a high reduction rate. Since the results of our experiments show promise, a next step would be to obtain a symbolic version of our algorithm and implement it inside a model-checking tool to determine if the reductions translate into faster running times for model-checking algorithms.

## Acknowledgements

This work was supported by the ERC Starting grant 279499 (inVEST) and grants from AFRL #FA8650-15-C-2546, DARPA #W911NF-16-1-0001, and ARO #W911NF-15-1-0592. G. A. Pérez is an F.R.S.-FNRS Aspirant and FWA post-doc fellow.

## References


[Aspnes and Herlihy, 1990] James Aspnes and Maurice Herlihy. Fast randomized consensus using shared memory. *J. of Algorithms*, 11(3):441–461, September 1990.

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

[Bernstein *et al.*, 2002] Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of Markov decision processes. *Math. Oper. Res.*, 27(4):819–840, November 2002.

[Brafman and Tennenholtz, 2003] Ronen I. Brafman and Moshe Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *J. Mach. Learn. Res.*, 3:213–231, March 2003.

[Brázdil *et al.*, 2014] Tomás Brázdil, Krishnendu Chatterjee, Martin Chmelik, Vojtech Forejt, Jan Kretínský, Marta Z. Kwiatkowska, David Parker, and Mateusz Ujma. Verification of Markov decision processes using learning algorithms. In *ATVA*, volume 8837 of *LNCS*, pages 98–114, 2014.

[Ciesinski *et al.*, 2008] Frank Ciesinski, Christel Baier, Marcus Größer, and Joachim Klein. Reduction techniques for model checking Markov decision processes. In *QEST*, pages 45–54, 2008.

[Condon, 1992] Anne Condon. The complexity of stochastic games. *Inf. Comput.*, 96(2):203–224, 1992.

[Courcoubetis and Yannakakis, 1995] Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42(4):857–907, 1995.

[De Alfaro, 1997] Luca De Alfaro. *Formal verification of probabilistic systems*. PhD thesis, Stanford University, 1997.

[Dehnert *et al.*, 2017] Christian Dehnert, Sebastian Junges, Joost-Pieter Katoen, and Matthias Volk. A storm is coming: A modern probabilistic model checker. In *CAV*, 2017.

[Ding *et al.*, 2014] X. Ding, S. L. Smith, C. Belta, and D. Rus. Optimal control of Markov decision processes with linear temporal logic constraints. *IEEE Trans. on Automatic Control*, 59(5):1244–1257, May 2014.

[Fu and Topcu, 2014] Jie Fu and Ufuk Topcu. Probably approximately correct MDP learning and control with temporal logic constraints. In *RSS*, 2014.

[Haddad and Monmege, 2014] Serge Haddad and Benjamin Monmege. Reachability in mdps: Refining convergence of value iteration. In Joël Ouaknine, Igor Potapov, and James Worrell, editors, *RP*, volume 8762 of *LNCS*, pages 125–137. Springer, 2014.

[Kaelbling *et al.*, 1996] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *JAIR*, 4:237–285, 1996.

[Kawaguchi, 2016] Kenji Kawaguchi. Bounded optimal exploration in MDP. In *AAAI*, pages 1758–1764, 2016.

[Kearns and Singh, 2002] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2):209–232, 2002.

[Kolobov *et al.*, 2011] Andrey Kolobov, Mausam, Daniel S. Weld, and Hector Geffner. Heuristic search for generalized stochastic shortest path mdps. In Fahiem Bacchus, Carmel Domshlak, Stefan Edelkamp, and Malte Helmert, editors, *ICAPS*. AAAI, 2011.

[Kolter and Ng, 2009] J. Z. Kolter and Andrew Y. Ng. Near-bayesian exploration in polynomial time. In Andrea P. Danyluk, Lon Bottou, and Michael L. Littman, editors, *ICML*, page 65, 2009.

[Kwiatkowska *et al.*, 2001] Marta Z. Kwiatkowska, Gethin Norman, and Roberto Segala. Automated verification of a randomized distributed consensus protocol using Cadence SMV and PRISM. In *CAV*, pages 194–206, 2001.

[Kwiatkowska *et al.*, 2004] Marta Z. Kwiatkowska, Gethin Norman, David Parker, and Jeremy Sproston. Performance analysis of probabilistic timed automata using digital clocks. In *FORMATS*, pages 105–120, 2004.

[Kwiatkowska *et al.*, 2011] Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of probabilistic real-time systems. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *CAV*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.

[Lacerda *et al.*, 2015] Bruno Lacerda, David Parker, and Nick Hawes. Optimal policy generation for partially satisfiable co-safe LTL specifications. In *IJCAI*, pages 1587–1593, 2015.

[Lahijanian *et al.*, 2010] M. Lahijanian, J. Wasniewski, S. B. Andersson, and C. Belta. Motion planning and control from temporal logic specifications with probabilistic satisfaction guarantees. In *ICRA*, pages 3227–3232, 2010.

[Littman, 1996] Michael L. Littman. *Algorithms for Sequential Decision Making*. PhD thesis, Brown University, 1996.

[Papadimitriou and Tsitsiklis, 1987] Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of Markov Decision Processes. *Math. Oper. Res.*, 12:441–450, 1987.

[Puterman, 2005] Martin L. Puterman. *Markov Decision Processes*. Wiley-Interscience, 2005.

[Russell and Norvig, 2010] Stuart J. Russell and Peter Norvig. *Artificial Intelligence - A Modern Approach (3. internat. ed.)*. Pearson Education, 2010.

[Russell *et al.*, 2015] Stuart J. Russell, Daniel Dewey, and Max Tegmark. Research priorities for robust and beneficial artificial intelligence. *AI Magazine*, 36(4), 2015.

[Steinmetz *et al.*, 2016] Marcel Steinmetz, Jörg Hoffmann, and Olivier Buffet. Goal probability analysis in probabilistic planning: Exploring and enhancing the state of the art. *JAIR*, 57:229–271, 2016.

[Strehl *et al.*, 2009] Alexander L. Strehl, Lihong Li, and Michael L. Littman. Reinforcement learning in finite MDPs: PAC analysis. *J. Mach. Learn. Res.*, 10:2413–2444, 2009.

[Svorenová *et al.*, 2013] María Svorenová, Ivana Cerna, and Calin Belta. Optimal control of MDPs with temporal logic constraints. In *CDC*, pages 3938–3943, 2013.

[Valiant, 2013] Leslie Valiant. *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*. Basic Books, 2013.

[Wen and Topcu, 2016] Min Wen and Ufuk Topcu. Probably approximately correct learning in stochastic games with temporal logic specifications. In Subbarao Kambhampati, editor, *IJCAI*, pages 3630–3636, 2016.