ACM Transaction on Modeling and Computer Simulation (TOMACS), PREPRINT VERSION, ACCEPTED, 2019

Interval Markov Decision Processes with Multiple Objectives: from Robust Strategies to Pareto Curves [*]

ERNST MORITZ HAHN, Department of Computer Science, University of Liverpool, UK
VAHID HASHEMI, Department of Information Technology, Audi AG, Germany
HOLGER HERMANNS, Saarland University, Saarland Informatics Campus, Germany
MORTEZA LAHIJANIAN, Department of Smead Aerospace Engineering and Sciences, University of Colorado, USA
ANDREA TURRINI, Institute of Intelligent Software, Guangzhou, China and State Key Laboratory of Computer Science, Institute of Software, CAS, China

Accurate Modelling of a real world system with probabilistic behaviour is a difficult task. Sensor noise and statistical estimations, among other imprecisions, make the exact probability values impossible to obtain. In this paper, we consider the Interval Markov decision processes (*IMDP*s), which generalise classical *MDP*s by having interval-valued transition probabilities. They provide a powerful modelling tool for probabilistic systems with an additional variation or uncertainty that prevents the knowledge of the exact transition probabilities. We investigate the problem of robust multi-objective synthesis for *IMDP*s and Pareto curve analysis of multi-objective queries on *IMDP*s. We study how to find a robust (randomised) strategy that satisfies multiple objectives involving rewards, reachability, and more general ω-regular properties against all possible resolutions of the transition probability uncertainties, as well as to generate an approximate Pareto curve providing an explicit view of the trade-offs between multiple objectives. We show that the multi-objective synthesis problem is **PSPACE**-hard and provide a value iteration-based decision algorithm to approximate the Pareto set of achievable points. We finally demonstrate the practical effectiveness of our proposed approaches by applying them on several case studies using a prototype tool.

CCS Concepts: • **Computing methodologies → Planning under uncertainty**; **Motion planning**; • **Theory of computation → Approximation algorithms analysis**;

Additional Key Words and Phrases: Interval Markov Decision Processes, Multi-objective Optimisation, Robust Synthesis, Pareto Curves, Complexity

[*]This work is supported by the ERC Advanced Investigators Grant 695614 (POWVER), by EPSRC Mobile Autonomy Programme Grant EP/M019918/1, by the CAS/SAFEA International Partnership Program for Creative Research Teams, by the National Natural Science Foundation of China (Grants No. 61550110506 and 61650410658), by the Chinese Academy of Sciences Fellowship for International Young Scientists, by H2020 Marie Skłodowska-Curie Actions Individual Fellowship "PaVeCo" - Parametrised Verification and Control, and by the CDZ project CAP (GZ 1023).

Authors' addresses: Ernst Moritz Hahn, Department of Computer Science, University of Liverpool, Liverpool, UK, e.m.hahn@liverpool.ac.uk; Vahid Hashemi, Department of Information Technology, Audi AG, Ingolstadt, Germany, vahid.hashemi@audi.de; Holger Hermanns, Saarland University, Saarland Informatics Campus, Saarbrücken, Germany, hermanns@cs.uni-saarland.de; Morteza Lahijanian, Department of Smead Aerospace Engineering and Sciences, University of Colorado, Boulder, CO, USA, morteza.lahijanian@colorado.edu; Andrea Turrini, Institute of Intelligent Software, Guangzhou, Guangzhou, China, State Key Laboratory of Computer Science, Institute of Software, CAS, Beijing, China, turrini@ios.ac.cn.

Manuscript submitted to ACM

ACM Reference Format:
Ernst Moritz Hahn, Vahid Hashemi, Holger Hermanns, Morteza Lahijanian, and Andrea Turrini. 2010. Interval Markov Decision Processes with Multiple Objectives: from Robust Strategies to Pareto Curves . *ACM Comput. Entertain.* 9, 4, Article 39 (March 2010), 31 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

Interval Markov Decision Processes (*IMDPs*) [Givan et al. 2000] extend classical *Markov Decision Processes* (*MDPs*) [Bellman 1957] by including uncertainty over the transition probabilities. More precisely, instead of a single value for the probability of taking a transition, *IMDPs* allow ranges of possible probability values given as closed intervals of the reals. Thereby, *IMDPs* provide a powerful modelling tool for probabilistic systems with an additional variation or uncertainty concerning the knowledge of exact transition probabilities. They are especially useful to represent realistic stochastic systems that, for instance, evolve in unknown environments with bounded behaviour or do not preserve the Markov property.

Since their introduction (under the name of bounded-parameter *MDPs*) [Givan et al. 2000], *IMDPs* have been receiving a lot of attention in the formal verification community [Cubuktepe et al. 2017; Petrucci and van de Pol 2018; Quatmann et al. 2016]. They are viewed as the appropriate abstraction model for uncertain systems with large state spaces, including continuous dynamical systems, for the purpose of analysis, verification, and control synthesis. Several model checking and control synthesis techniques have been developed [Puggelli 2014; Puggelli et al. 2013; Wolff et al. 2012] causing a boost in the applications of *IMDPs*, ranging from verification of continuous stochastic systems (e.g., [Lahijanian et al. 2015]) to robust strategy synthesis for robotic systems (e.g., [Luna et al. 2014a,b,c; Wolff et al. 2012]).

In recent years, there has been an increasing interest in multi-objective strategy synthesis for probabilistic systems [Chatterjee et al. 2006; Esteve et al. 2012; Forejt et al. 2011, 2012; Kwiatkowska et al. 2013; Mouaddib 2004; Ogryczak et al. 2013; Perny et al. 2013; Randour et al. 2015]. Here, the goal is first to provide a complete trade-off analysis of several, possibly conflicting, quantitative properties and then to synthesise a strategy that guarantees the user's desired behaviour. Such properties, for instance, ask to "find a robot strategy that maximises p_{safe}, the probability of successfully completing a track by safely manoeuvring between obstacles, while minimising t_{travel}, the total expected travel time". This example has competing objectives: maximising p_{safe}, which requires the robot to be conservative, and minimising t_{travel}, which causes the robot to be reckless. In such contexts, the interest is in the *Pareto curve* of the possible solution points: the set of all pairs of $(p_{\text{safe}}, t_{\text{travel}})$ for which an increase in the value of p_{safe} must induce an increase in the value of t_{travel}, and vice versa. Given a point on the curve, the computation of the corresponding strategy is asked.

Existing multi-objective synthesis frameworks [Chatterjee et al. 2006; Esteve et al. 2012; Forejt et al. 2011, 2012; Kwiatkowska et al. 2013; Mouaddib 2004; Ogryczak et al. 2013; Perny et al. 2013; Randour et al. 2015] are limited to *MDP* models of probabilistic systems. The algorithms use iterative methods (similar to value iteration) for the computation of the Pareto curve and rely on reductions to linear programming for strategy synthesis. As discussed above, *MDPs*, however, are constrained to single-valued transition probabilities, posing severe limitations for many real-world systems.

In this paper, we present novel techniques for robust control of *IMDPs* with multiple objectives. Our aim is to approximate Pareto curve for a set of conflicting objectives, despite the additional uncertainty over the transition probabilities in these models. Our approach views the uncertainty as making adversarial choices among the available transition probability distributions induced by the intervals, as the system evolves. This is contract to works like [Scheftelowitsch et al. 2017] where a probability distribution about the intervals is assumed and similar approaches [Petrucci and van de Pol 2018]. We refer to this as the *controller synthesis* semantics. We compute a successive and increasingly precise approximation of the Pareto curve through a value iteration algorithm which optimises the weighted sum of objectives. We consider three different multi-objective queries for *IMDP*s, namely synthesis, quantitative, and Pareto queries. We start with the synthesis queries where our goal is to synthesise a robust strategy that guarantees the satisfaction of a multi-objective property. We first analyse the problem complexity and prove that it is **PSPACE**-hard and then develop a value iteration-based algorithm to approximate the Pareto curve of the given set of objectives. Afterwards, we extend our solution approach to approximate the Pareto curve for other types of queries. In order to show the effectiveness of our approach, we present promising results on several case studies analysed by a prototype implementation of the algorithms.

Our queries are formulated in a way similar to [Forejt et al. 2012] but with three key extensions. First of all, we discuss approximating Pareto curves for *IMDP* models which include interval model of uncertainty and provide more expressive modelling formalisms for the abstraction of real world systems. As we discuss later, our solution approach can also handle *MDP* models with more general convex models of uncertainty. Next, we provide a detailed discussion on the reduction of a multi-objective property including reachability or reward predicates to a basic form, i.e., a multi-objective property including only reward predicates. Our reduction to the basic form extends its counterpart in [Forejt et al. 2011, 2012] for *MDP*s. It also corrects a few minor flaws of these works, in particular in [Forejt et al. 2012, Proposition 2]; see the discussion after Proposition 18.

Finally, we detail the generation of randomised strategies.

This article is an extended version of [Hahn et al. 2017]; compared with [Hahn et al. 2017], in this paper we provide additional technical details such as formal proofs, the extension to general PLTL and ω-regular properties, the generation of randomised strategies, and additional empirical results.

Related work. Related work can be grouped into two main categories: uncertain Markov model formalisms and model checking/synthesis algorithms.

Firstly, from the modelling viewpoint, various probabilistic modelling formalisms with uncertain transitions have been studied in the literature. Interval Markov Chains (*IMC*s) [Jonsson and Larsen 1991; Kozine and Utkin 2002] or abstract Markov chains [Fecher et al. 2006] extend standard discrete-time Markov Chains (*MC*s) with interval uncertainties. They do not feature the nondeterministic choices of transitions. Uncertain *MDP*s [Puggelli et al. 2013] allow more general sets of distributions to be associated with each transition, not only those described by intervals. They usually are restricted to *rectangular uncertainty sets* requiring that the uncertainty is linear and independent for any two transitions of any two states. Parametric *MDP*s [Daws 2004; Hahn et al. 2011], to the contrary, allow such dependencies as every probability is described as a rational function on a finite set of global parameters. *IMDP*s extend *IMC*s by inclusion of nondeterminism and are a subset of uncertain *MDP*s and parametric *MDP*s.

Secondly, from the side of algorithmic developments, several verification methods for uncertain Markov models have been proposed. The problem of computing reachability probabilities and expected total reward for *IMC*s and *IMDP*s was first investigated in [Chen et al. 2013b; Wu and Koutsoukos 2008]. Then, several of PCTL and LTL model checking algorithms discussed in these works were introduced in [Benedikt et al. 2013; Chatterjee et al. 2008; Chen et al. 2013b] and [Lahijanian et al. 2015; Puggelli et al. 2013; Wolff et al. 2012], respectively. Concerning strategy synthesis algorithms, the works in [Hahn et al. 2011; Nilim and El Ghaoui 2005] considered synthesis for parametric *MDP*s and *MDP*s with ellipsoidal uncertainty in the verification community. In control community, such synthesis problems were mostly studied for uncertain Markov models in [Givan et al. 2000; Nilim and El Ghaoui 2005; Wu and Koutsoukos 2008] with the aim to maximise expected finite-horizon (un)discounted rewards. All these works, however, consider solely single objective properties, and their extension to multi-objective synthesis is not trivial.

Multi-objective model checking of probabilistic models with respect to various quantitative objectives has been recently investigated. The works of [Etessami et al. 2007; Forejt et al. 2011, 2012; Kwiatkowska et al. 2013] focused on multi-objective verification of ordinary *MDP*s. In [Chen et al. 2013a], these algorithms were extended to the more general models of 2-player stochastic games. These models, however, cannot capture the continuous uncertainty in the transition probabilities as *IMDP*s do. For the purposes of synthesis though, it is possible to transform an *IMDP* into a 2-player stochastic game; nevertheless, such a transformation raises an extra exponential factor to the complexity of the decision problem. This exponential blowup has been avoided in our setting.

Structure of the paper. We start with necessary preliminaries in Section 2. In Section 3, we discuss multi-objective robust control of *IMDP*s and present our novel solution approaches. In Section 4, we detail how randomised strategies can be generated. In Section 5, we demonstrate our approach on three case studies and present experimental results. Finally, in Section 6 we conclude the paper.

To keep the presentation clear, non-trivial proofs have been moved to the Appendix A.

2 PRELIMINARIES

For a set X, denote by $\mathrm{Disc}(X)$ the sets of discrete probability distributions over X. A discrete probability distribution ρ is a function $\rho\colon X \to \mathbb{R}_{\geq 0}$ such that $\sum_{x \in X} \rho(x) = 1$; for $X' \subseteq X$, we write $\rho(X')$ for $\sum_{x \in X'} \rho(x)$. Given $\rho \in \mathrm{Disc}(X)$, we denote by $\mathrm{Supp}(\rho)$ the set $\{\, x \in X \mid \rho(x) > 0 \,\}$, and by δ_x, where $x \in X$, the *point* distribution such that $\delta_x(y) = 1$ for $y = x$, 0 otherwise. For a distribution ρ, we also write $\rho = \{\, (x, p_x) \mid x \in X \,\}$ where $p_x = \rho(x)$ is the probability of x.

For a vector $\mathbf{x} \in \mathbb{R}^n$ we denote by x_i, its i-th component, and we call $\mathbf{x}$ a *weight vector* if $x_i \geq 0$ for all i and $\sum_{i=1}^{n} x_i = 1$. The Euclidean inner product $\mathbf{x} \cdot \mathbf{y}$ of two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ is defined as $\sum_{i=1}^{n} x_i \cdot y_i$. In the following, when comparing vectors, the comparison is to be understood component-wise. Thus, e.g. $\mathbf{x} \leq \mathbf{y}$ means that for all indices i we have $x_i \leq y_i$. For a set of vectors $S = \{\mathbf{s}_1, \dots, \mathbf{s}_t\} \subseteq \mathbb{R}^n$, we say that $\mathbf{s} \in \mathbb{R}^n$ is a *convex combination* of elements of S, if $\mathbf{s} = \sum_{i=1}^{t} w_i \cdot \mathbf{s}_i$ for some weight vector $\mathbf{w} \in \mathbb{R}^t_{\geq 0}$. Furthermore, we denote by $S\!\downarrow$ the *downward closure* of the convex hull of S which is defined as $S\!\downarrow = \{\, \mathbf{y} \in \mathbb{R}^n \mid \mathbf{y} \leq \mathbf{z}$ for some convex combination $\mathbf{z}$ of the elements of $S \,\}$. For a given convex set X, we say that a point $\mathbf{x} \in X$ is on the boundary of X, denoted by $\mathbf{x} \in \partial X$, if for every $\varepsilon > 0$ there is a point $\mathbf{y} \notin X$ such that the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$ is at most ε. Given a downward closed set $X \in \mathbb{R}^n$, for any $\mathbf{z} \in \mathbb{R}^n$ such that $\mathbf{z} \in \partial X$ or $\mathbf{z} \notin X$, there is a weight vector $\mathbf{w} \in \mathbb{R}^n$ such that $\mathbf{w} \cdot \mathbf{z} \geq \mathbf{w} \cdot \mathbf{x}$ for all $\mathbf{x} \in X$ [Boyd and Vandenberghe 2004]. We say that $\mathbf{w}$ separates $\mathbf{z}$ from $X\!\downarrow$. Given a set $Y \subseteq \mathbb{R}^k$, we call a vector $\mathbf{y} \in Y$ *Pareto optimal* in Y if there does not exist a vector $\mathbf{z} \in Y$ such that $\mathbf{y} \leq \mathbf{z}$ and $\mathbf{y} \neq \mathbf{z}$. We define the *Pareto set* or *Pareto curve* of Y to be the set of all Pareto optimal vectors in Y, i.e., Pareto set $\mathcal{Y} = \{\, \mathbf{y} \in Y \mid \mathbf{y}$ is Pareto optimal $\}$.

2.1 Interval Markov Decision Processes

We now define *Interval Markov Decision Processes* (*IMDP*s) as an extension of *MDP*s, which allows for the inclusion of transition probability uncertainties as *intervals*. *IMDP*s belong to the family of uncertain *MDP*s and allow to describe a set of *MDP*s with identical (graph) structures that differ in distributions associated with transitions. Formally,

Definition 1 (IMDPs). An *Interval Markov Decision Process* (*IMDP*) $\mathcal{M}$ is a tuple $(S, \bar{s}, \mathcal{A}, I, AP, L)$, where S is a finite set of *states*, $\bar{s} \in S$ is the initial state, $\mathcal{A}$ is a finite set of *actions*, $I\colon S \times \mathcal{A} \times S \to \mathbb{I} \cup \{[0, 0]\}$ is a total *interval transition* *probability function where* $\mathbb{I} = \{ [a, b] \mid 0 < a \leq b \leq 1 \}$, AP if a finite set of *atomic propositions*, and $L\colon S \to 2^{AP}$ is a total labelling function.

The requirement that $0 < a$ ensures that the graph structure remains the same for different resolutions of the intervals. Having $a = 0$ would mean that an edge in the graph could disappear. As discussed later on, this restriction is essential for some of the algorithms we use to analyse *IMDP*s. Given $s \in S$ and $a \in \mathcal{A}$, we call $\mathfrak{h}^a_s \in \mathrm{Disc}(S)$ a *feasible distribution* reachable from s by a, denoted by $s \xrightarrow{a} \mathfrak{h}^a_s$, if, for each state $s' \in S$, we have $\mathfrak{h}^a_s(s') \in I(s, a, s')$. This means that we can only assign probability values lying in the interval $I(s, a, s')$ to state s'. We denote the set of feasible distributions for state s and action a by $\mathcal{H}^a_s$, i.e., $\mathcal{H}^a_s = \{ \mathfrak{h}^a_s \in \mathrm{Disc}(S) \mid s \xrightarrow{a} \mathfrak{h}^a_s \}$ and we denote the set of available actions at state $s \in S$ by $\mathcal{A}(s)$, i.e., $\mathcal{A}(s) = \{ a \in \mathcal{A} \mid \mathcal{H}^a_s \neq \emptyset \}$. We assume that $\mathcal{A}(s) \neq \emptyset$ for all $s \in S$. We define the *size* of $\mathcal{M}$, written $|\mathcal{M}|$, as the number of non-zero entries of I, i.e., $|\mathcal{M}| = |\{ (s, a, s', \iota) \in S \times \mathcal{A} \times S \times \mathbb{I} \mid I(s, a, s') = \iota \}| \in O(|S|^2 \cdot |\mathcal{A}|)$.

A *path* ξ in $\mathcal{M}$ is a finite or infinite sequence of alternating states and actions $\xi = s_0 a_0 s_1 \dots$, ending with a state if finite, such that for each $i \geq 0$, $I(s_i, a_i, s_{i+1}) \in \mathbb{I}$. The i-th state (action) along the path ξ is denoted by $\xi[i]$ ($\xi(i)$) and, if the path is finite, we denote by $last(\xi)$ its last state; moreover, we denote by $\xi[i \dots]$ the suffix of ξ starting from $\xi[i]$. For instance, for the finite path $\xi = s_0 a_0 s_1 \dots s_n$, we have $\xi[i] = s_i$, $\xi(i) = a_i$, and $last(\xi) = s_n$. The sets of all finite and infinite paths in $\mathcal{M}$ are denoted by *FPaths* and *IPaths*, respectively.

An ω*-word* w is an infinite sequence of sets of atomic propositions, i.e., $w \in (2^{AP})^\omega$. Given an infinite path ξ, the word $w(\xi)$ generated by ξ is the sequence $w(\xi) = w_0 w_1 \dots$ such that for each $i \geq 0$, $w_i = L(\xi[i])$.

The nondeterministic choices between available actions and feasible distributions present in an *IMDP* are resolved by strategies and natures, respectively.

Definition 2 (Strategy and Nature in IMDPs). Given an *IMDP* $\mathcal{M}$, a *strategy* is a function $\sigma\colon FPaths \to \mathrm{Disc}(\mathcal{A})$ such that for each $\xi \in FPaths$, $\sigma(\xi) \in \mathrm{Disc}(\mathcal{A}(last(\xi))$. A *nature* is a function $\pi\colon FPaths \times \mathcal{A} \to \mathrm{Disc}(S)$ such that for each $\xi \in FPaths$ and $a \in \mathcal{A}(s)$, $\pi(\xi, a) \in \mathcal{H}^a_s$ where $s = last(\xi)$. The sets of all strategies and all natures are denoted by Σ and Π, respectively.

Given a finite path ξ of an *IMDP*, a strategy σ, and a nature π, the system evolution proceeds as follows: let $s = last(\xi)$. First, an action $a \in \mathcal{A}(s)$ is chosen probabilistically by σ. Then, π resolves the uncertainties and chooses one feasible distribution $\mathfrak{h}^a_s \in \mathcal{H}^a_s$. Finally, the next state s' is chosen according to the distribution $\mathfrak{h}^a_s$, and the path ξ is extended by a and s', i.e., the resulting path is $\xi' = \xi a s'$.

A strategy σ and a nature π induce a probability measure over paths as follows. The basic measurable events are the cylinder sets of finite paths, where the *cylinder set* of a finite path ξ is the set $Cyl_\xi = \{ \xi' \in IPaths \mid \xi \text{ is a prefix of } \xi' \}$. The probability $\Pr^{\sigma,\pi}_{\mathcal{M}}$ of a cylinder set Cyl_ξ is defined inductively as follows:

$$\Pr^{\sigma,\pi}_{\mathcal{M}}(Cyl_\xi) = \begin{cases} 1 & \text{if } \xi = \bar{s}, \\ 0 & \text{if } \xi = t \neq \bar{s}, \\ \Pr^{\sigma,\pi}_{\mathcal{M}}(Cyl_{\xi'}) \cdot \sigma(\xi')(a) \cdot \pi(\xi', a)(s) & \text{if } \xi = \xi' a s. \end{cases}$$

Standard measure theoretical arguments ensure that $\Pr^{\sigma,\pi}_{\mathcal{M}}$ extends uniquely to the σ-field generated by cylinder sets.

In order to model additional quantitative measures of an *IMDP*, we associate rewards to the enabled actions. This is done by means of *reward structures*.

Definition 3 (Reward Structure). A *reward structure* for an *IMDP* is a function $\mathsf{r}\colon S \times \mathcal{A} \to \mathbb{R}$ that assigns to each state-action pair (s, a), where $s \in S$ and $a \in \mathcal{A}(s)$, a reward $\mathsf{r}(s, a) \in \mathbb{R}$. Given a path ξ and $k \in \mathbb{N} \cup \{\infty\}$, the total accumulated reward in k steps for ξ over r is $\mathsf{r}[k](\xi) = \sum_{i=0}^{k-1} \mathsf{r}(\xi[i], \xi(i))$.

Note that we allow negative rewards in this definition; however, due to later assumptions, their use is restricted. In particular, negative rewards are only allowed as result of the encoding of probability values as specified in Proposition 18.

Example 4. As an example of *IMDP* with a reward structure, consider the *IMDP* depicted in Fig. 1. The set of states is $S = \{s, t, u\}$ with s being the initial one. The set of actions is $\mathcal{A} = \{a, b\}$, and the non-zero transition probability intervals are $I(s, a, t) = [\frac{1}{3}, \frac{2}{3}]$, $I(s, a, u) = [\frac{1}{10}, 1]$, $I(s, b, t) = [\frac{2}{5}, \frac{3}{5}]$, $I(s, b, u) = [\frac{1}{4}, \frac{2}{3}]$, and $I(t, a, t) = I(u, b, u) = [1, 1]$. The underlined numbers indicate the reward structure r with $\mathsf{r}(s, a) = 3$, $\mathsf{r}(s, b) = 1$, and $\mathsf{r}(t, a) = \mathsf{r}(u, b) = 0$. Among the uncountable many distributions belonging to $\mathcal{H}_s^a$, two possible choices for nature π on s and a are $\pi(s, a) = \{(t, \frac{3}{5}), (u, \frac{2}{5})\}$ and $\pi(s, a) = \{(t, \frac{1}{3}), (u, \frac{2}{3})\}$. $\diamondsuit$

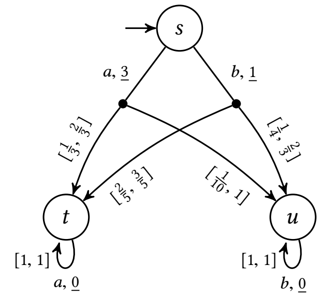


Fig. 1. An example of *IMDP*.

2.2 Probabilistic Linear Time Logic (PLTL)

Probabilistic Linear Time Logic (PLTL) [Bianco and de Alfaro 1995] is the probabilistic counterpart of LTL for Kripke structures which can be used to express properties of an IMDP with respect to its infinite behaviour, such as liveness properties. Let *AP* be a given set of atomic propositions. The syntax of a PLTL formula Φ is given by:

$$\Phi ::= Pr_{\sim p}[\Psi] \mid Pr_{\min=?}[\Psi] \mid Pr_{\max=?}[\Psi]$$
$$\Psi ::= a \mid \neg\Psi \mid \Psi \wedge \Psi \mid \mathbf{X}\Psi \mid \Psi\,\mathbf{U}\,\Psi$$

where $a \in AP$, $\sim \in \{\leq, \geq\}$, and $p \in [0, 1] \cap \mathbb{Q}$. Standard Boolean operators such as false, true, disjunction, implication, equivalence can be derived as usual, e.g., $f\!f = a \wedge \neg a$, $tt = \neg f\!f$, and $\Psi_1 \vee \Psi_2 = \neg(\neg\Psi_1 \wedge \neg\Psi_2)$; similarly, the finally $\mathbf{F}$ and globally $\mathbf{G}$ temporal operators can be defined as $\mathbf{F}\Psi = tt\,\mathbf{U}\,\Psi$ and $\mathbf{G}\Psi = \neg\mathbf{F}\neg\Psi$.

Note that a PLTL formula Φ is just a probability operator on top of an LTL formula Ψ; this is clear by the semantics of Φ and Ψ: given an *IMDP* $\mathcal{M}$ and a PLTL formula $Pr_{\sim p}[\Psi]$, we say that $\mathcal{M}$ satisfies $Pr_{\sim p}[\Psi]$, written $\mathcal{M} \models Pr_{\sim p}[\Psi]$, if $\Pr_{\mathcal{M}}^{\sigma,\pi}(\{\, \xi \in IPaths \mid \xi \models \Psi \,\}) \sim p$ for all $\sigma \in \Sigma$ and $\pi \in \Pi$, where $\xi \models \Psi$ is defined inductively as follows:

$$
\begin{array}{ll}
\xi \models a & \text{if } a \in L(\xi[0]), \\
\xi \models \neg\Psi & \text{if it is not the case that } \xi \models \Psi \text{ (also written } \xi \not\models \Psi), \\
\xi \models \Psi_1 \wedge \Psi_2 & \text{if } \xi \models \Psi_1 \text{ and } \xi \models \Psi_2, \\
\xi \models \mathbf{X}\Psi & \text{if } \xi[1 \ldots] \models \Psi, \text{ and} \\
\xi \models \Psi_1\,\mathbf{U}\,\Psi_2 & \text{if there exists } n \in \mathbb{N} \text{ such that } \xi[n \ldots] \models \Psi_2 \text{ and for each } 0 \leq i < n, \text{ it holds } \xi[i \ldots] \models \Psi_1.
\end{array}
$$

The value of the PLTL formula $Pr_{\mathrm{opt}=?}[\Psi]$, with $\mathrm{opt} \in \{\min, \max\}$, is defined as

$$Pr_{\mathrm{opt}=?}[\Psi] = \operatorname*{opt}_{\sigma \in \Sigma, \pi \in \Pi} \Pr_{\mathcal{M}}^{\sigma,\pi}(\{\, \xi \in IPaths \mid \xi \models \Psi \,\}).$$

3 MULTI-OBJECTIVE ROBUST CONTROL OF IMDPs

In this section, we start by considering two main classes of properties for *IMDP*s; the *probability of reaching a target* and the *expected total reward*. The reason that we focus on these properties is that their algorithms usually serve as the basis for more complex properties, such as quantitative properties and PLTL/ω-regular properties, as we will present later in the section. To this aim, we lift the satisfaction definition of these two classes of properties from *MDPs* [Forejt et al. 2011, 2012] to *IMDPs* by encoding the notion of robustness for strategies.

Definition 5 (Reachability Predicate & its Robust Satisfaction). A *reachability predicate* $[T]_{\sim p}^{\leq k}$ consists of a set of target states $T \subseteq S$, a relational operator $\sim \in \{\leq, \geq\}$, a rational probability bound $p \in [0,1] \cap \mathbb{Q}$ and a time bound $k \in \mathbb{N} \cup \{\infty\}$. It indicates that the probability of reaching T within k time steps satisfies $\sim p$.

Robust satisfaction of $[T]_{\sim p}^{\leq k}$ by *IMDP* $\mathcal{M}$ under strategy $\sigma \in \Sigma$ is denoted by $\mathcal{M}\downharpoonright_{\sigma} \models_{\Pi} [T]_{\sim p}^{\leq k}$ and indicates that the probability of the set of all paths that reach T under σ satisfies the bound $\sim p$ for every choice of nature $\pi \in \Pi$. Formally, $\mathcal{M}\downharpoonright_{\sigma} \models_{\Pi} [T]_{\sim p}^{\leq k}$ iff $\Pr_{\mathcal{M}}^{\sigma}(\Diamond^{\leq k} T) \sim p$ where $\Pr_{\mathcal{M}}^{\sigma}(\Diamond^{\leq k} T) = \mathrm{opt}_{\pi \in \Pi} \Pr_{\mathcal{M}}^{\sigma,\pi}(\{\xi \in \mathit{IPaths} \mid \exists i \leq k : \xi[i] \in T\})$ and $\mathrm{opt} = \min$ if $\sim\, = \,\geq$ and $\mathrm{opt} = \max$ if $\sim\, = \,\leq$. Furthermore, σ is referred to as a robust strategy.

Definition 6 (Reward Predicate & its Robust Satisfaction). A *reward predicate* $[\mathsf{r}]_{\sim r}^{\leq k}$ consists of a reward structure r, a time bound $k \in \mathbb{N} \cup \{\infty\}$, a relational operator $\sim \in \{\leq, \geq\}$ and a reward bound $r \in \mathbb{Q}$. It indicates that the expected total accumulated reward within k steps satisfies $\sim r$.

Robust satisfaction of $[\mathsf{r}]_{\sim r}^{\leq k}$ by *IMDP* $\mathcal{M}$ under strategy $\sigma \in \Sigma$ is denoted by $\mathcal{M}\downharpoonright_{\sigma} \models_{\Pi} [\mathsf{r}]_{\sim r}^{\leq k}$ and indicates that the expected total reward over the set of all paths under σ satisfies the bound $\sim r$ for every choice of nature $\pi \in \Pi$. Formally, $\mathcal{M}\downharpoonright_{\sigma} \models_{\Pi} [\mathsf{r}]_{\sim r}^{\leq k}$ iff $\mathit{ExpTot}_{\mathcal{M}}^{\sigma,k}[\mathsf{r}] \sim r$ where $\mathit{ExpTot}_{\mathcal{M}}^{\sigma,k}[\mathsf{r}] = \mathrm{opt}_{\pi \in \Pi} \int_{\xi} \mathsf{r}[k](\xi)\, \mathrm{d}\Pr_{\mathcal{M}}^{\sigma,\pi}$ and $\mathrm{opt} = \min$ if $\sim\, = \,\geq$ and $\mathrm{opt} = \max$ if $\sim\, = \,\leq$. Furthermore, σ is referred to as the robust strategy.

For the purpose of algorithm design, we also consider weighted sum of rewards. Formally,

Definition 7 (Weighted Reward Sum). Given a weight vector $\mathbf{w} \in \mathbb{R}^n$, a vector of time bounds $\mathbf{k} = (k_1, \ldots, k_n) \in (\mathbb{N} \cup \{\infty\})^n$ and reward structures $\mathsf{r} = (\mathsf{r}_1, \ldots, \mathsf{r}_n)$ for an *IMDP* $\mathcal{M}$, the *weighted reward sum* $\mathbf{w} \cdot \mathsf{r}[\mathbf{k}]$ over a path ξ is defined as $\mathbf{w} \cdot \mathsf{r}[\mathbf{k}](\xi) = \sum_{i=1}^{n} w_i \cdot \mathsf{r}_i[k](\xi)$. The *expected total weighted sum* is defined as $\mathit{ExpTot}_{\mathcal{M}}^{\sigma,\mathbf{k}}[\mathbf{w} \cdot \mathsf{r}] = \max_{\pi \in \Pi} \int_{\xi} \mathbf{w} \cdot \mathsf{r}[\mathbf{k}](\xi)\, \mathrm{d}\Pr_{\mathcal{M}}^{\sigma,\pi}$ for bounds $\leq$ and accordingly minimises over natures for $\geq$; for a given strategy σ, we have: $\mathit{ExpTot}_{\mathcal{M}}^{\sigma,\mathbf{k}}[\mathbf{w} \cdot \mathsf{r}] = \sum_{i=1}^{n} w_i \cdot \mathit{ExpTot}_{\mathcal{M}}^{\sigma,k_i}[\mathsf{r}_i]$.

3.1 Multi-objective Queries

Multi-objective properties for *IMDPs* essentially require multiple predicates to be satisfied at the same time under the same strategy for every choice of the nature. We now explain how to formalise multi-objective queries for *IMDPs*.

Definition 8 (Multi-objective Predicate). A *multi-objective predicate* is a vector $\varphi = (\varphi_1, \ldots, \varphi_n)$ of reachability or reward predicates. We say that φ is satisfied by *IMDP* $\mathcal{M}$ under strategy σ for every choice of nature $\pi \in \Pi$, denoted by $\mathcal{M}\downharpoonright_{\sigma} \models_{\Pi} \varphi$ if, for each $1 \leq i \leq n$, we have $\mathcal{M}\downharpoonright_{\sigma} \models_{\Pi} \varphi_i$. We refer to σ as a robust strategy. Furthermore, we call φ a basic multi-objective predicate if it is of the form $([\mathsf{r}_1]_{\geq r_1}^{\leq k_1}, \ldots, [\mathsf{r}_n]_{\geq r_n}^{\leq k_n})$, i.e., it includes only lower-bounded reward predicates.

We formulate multi-objective queries for *IMDPs* in three ways namely, *synthesis queries*, *quantitative queries*, and *Pareto queries*. We first formulate multi-objective synthesis queries for *IMDPs* as follows.

Definition 9 (Synthesis Query). Given an *IMDP* $\mathcal{M}$ and a multi-objective predicate φ, the *synthesis query* asks if there exists a robust strategy $\sigma \in \Sigma$ such that $\mathcal{M}\downharpoonright_{\sigma} \models_{\Pi} \varphi$.

Note that the synthesis queries check for the existence of a robust strategy that satisfies a multi-objective predicate φ for every resolution of nature.

The next type of queries is multi-objective quantitative queries which are defined as follows.

Definition 10 (Quantitative Query). Given an *IMDP* $\mathcal{M}$ and a multi-objective predicate φ, a quantitative query is of the form $qnt([o]^{\leq k_1}_{\mathrm{opt}}, (\varphi_2, \dots, \varphi_n))$, consisting of a multi-objective predicate $(\varphi_2, \dots, \varphi_n)$ of size $n-1$ and an objective $[o]^{\leq k_1}_{\mathrm{opt}}$ where o is a target set T or a reward structure r, $k_1 \in \mathbb{N} \cup \{\infty\}$ and $\mathrm{opt} \in \{\min, \max\}$. We define:

$$qnt([o]^{\leq k_1}_{\min}, (\varphi_2, \dots, \varphi_n)) = \inf\{\, x \in \mathbb{R} \mid ([o]^{\leq k_1}_{\leq x}, \varphi_2, \dots, \varphi_n) \text{ is satisfiable} \,\}$$

$$qnt([o]^{\leq k_1}_{\max}, (\varphi_2, \dots, \varphi_n)) = \sup\{\, x \in \mathbb{R} \mid ([o]^{\leq k_1}_{\geq x}, \varphi_2, \dots, \varphi_n) \text{ is satisfiable} \,\}.$$

Quantitative queries ask to maximise or minimise the reachability/reward objective over the set of strategies satisfying a given multi-objective predicate φ.

The last type of queries is multi-objective Pareto queries which ask to determine the Pareto set for a given set of objectives. Multi-objective Pareto queries are defined as follows.

Definition 11 (Pareto Query). Given an *IMDP* $\mathcal{M}$ and a multi-objective predicate φ, a Pareto query is of the form $Pareto([o_1]^{\leq k_1}_{\mathrm{opt}_1}, \dots, [o_n]^{\leq k_n}_{\mathrm{opt}_n})$, where each $[o_i]^{\leq k_i}_{\mathrm{opt}_i}$ is an objective in which o_i is either a target set T_i or a reward structure r_i, $k_i \in \mathbb{N} \cup \{\infty\}$, and $\mathrm{opt}_i \in \{\min, \max\}$. We define the set of achievable values as $A = \{\, \mathbf{x} \in \mathbb{R}^n \mid ([o_1]^{\leq k_1}_{\sim_1 x_1}, \dots, [o_n]^{\leq k_n}_{\sim_n x_n}) \text{ is satisfiable} \,\}$ where $\sim_i \,=\, \geq$ if $\mathrm{opt}_i = \max$, or $\sim_i \,=\, \leq$ if $\mathrm{opt}_i = \min$. Then,

$$Pareto([o_1]^{\leq k_1}_{\mathrm{opt}_1}, \dots, [o_n]^{\leq k_n}_{\mathrm{opt}_n}) = \{\, \mathbf{x} \in A \mid \mathbf{x} \text{ is Pareto optimal} \,\}.$$

There are some corner cases under which our proposed algorithms would not work correctly, such as for instance when the total expected reward could become infinite in a given model. Therefore, we need to limit the usage of rewards by assuming reward-finiteness for the strategies that satisfy the

Assumption 1 (Reward-finiteness). *Suppose that an IMDP $\mathcal{M}$ and a synthesis query φ are given. Let $\varphi = ([T_1]^{\leq k_1}_{\sim p_1}, \dots, [T_n]^{\leq k_n}_{\sim p_n}, [\mathsf{r}_{n+1}]^{\leq k_{n+1}}_{\sim r_{n+1}}, \dots, [\mathsf{r}_m]^{\leq k_m}_{\sim r_m})$. We say that φ is reward-finite if for each $n+1 \leq i \leq m$ such that $k_i = \infty$, we have $\sup_{\sigma \in \Sigma}\{\, ExpTot^{\sigma, k_i}_{\mathcal{M}}[\mathsf{r}_i] \mid \mathcal{M}\!\downharpoonright_\sigma \models_\Pi ([T_1]^{\leq k_1}_{\sim p_1}, \dots, [T_n]^{\leq k_n}_{\sim p_n}) \,\} < \infty$.*

In the next section we provide a method to check for reward-finiteness assumption of a given *IMDP* $\mathcal{M}$ and a synthesis query φ, a preprocessing procedure that removes actions with non-zero rewards from the end components of $\mathcal{M}$, and a proof for the correctness of this procedure with respect to φ. In the rest of the paper, we assume that all queries are reward-finite. Furthermore, for the soundness of our analysis we also require that for any *IMDP* $\mathcal{M}$ and φ given as in Assumption 1, the following properties hold: *(i)* each reward structure r_i assigns only non-negative values; *(ii)* φ is reward-finite; and *(iii)* for indices $n+1 \leq i \leq m$ such that $k_i = \infty$, either all $\sim_i$s are $\leq$ or all are $\geq$.

3.2 A Procedure to Check Assumption 1

In this section, we discuss in detail how reward-finiteness assumption for a given *IMDP* $\mathcal{M}$ and a synthesis query φ can be checked. Once it is known that the assumption is satisfied, the *IMDP* $\mathcal{M}$ can then be pruned to simplify the analysis. The idea underlying pruning is to remove transitions (and states) from the end-components that make the expected reward infinite under strategies not satisfying the reachability constraints in φ. In order to describe the procedure that checks Assumption 1, first we need to define a counterpart of end components of *MDP*s for *IMDP*s, to which we refer as a *strong end-component* (SEC). Intuitively, a SEC of an *IMDP* is a sub-*IMDP* for which there exists a strategy that forces the sub-*IMDP* to remain in the end component and visit all its states infinitely often under any nature. It is referred to as strong because it is independent of the choice of nature. Formally,

Definition 12 (Strong End-Component). A *strong end-component* (SEC) of an *IMDP* $\mathcal{M}$ is $E_{\mathcal{M}} = (S', \mathcal{A}')$, where $S' \subseteq S$ and $\mathcal{A}' \subseteq \bigcup_{s \in S'} \{s\} \times \mathcal{A}(s)$ such that (1) $\sum_{s' \in S'} \mathfrak{h}^a_{ss'} = 1$ for each $s \in S'$, $(s, a) \in \mathcal{A}'$, and $\mathfrak{h}^a_s \in \mathcal{H}^a_s$; and (2) for each $s, s' \in S'$ there is a finite path $\xi = \xi[0] \cdots \xi[n]$ such that $\xi[0] = s$, $\xi[n] = s'$, and for each $0 \le i \le n-1$ we have $\xi[i] \in S'$ and $(\xi[i], \xi(i)) \in \mathcal{A}'$.

Remark 13. *The SECs of an IMDP $\mathcal{M}$ can be identified by using any end-component-search algorithm of MDPs on its underlying graph structure. That is, since the lower transition probability bounds of $\mathcal{M}$ are strictly greater than zero for the transitions whose upper probability bounds are non-zero, the underlying graph structure of $\mathcal{M}$ is identical to the graph structure of every MDP it contains. Therefore, a SEC of $\mathcal{M}$ is an end-component of every contained MDP, and vice versa.*

Lemma 14. *If a state-action pair (s, a) is not contained in a SEC, then*

$$\sup_{\sigma \in \Sigma} \inf_{\pi \in \Pi} occ^{\sigma}_{\pi}(s, a) < \infty,$$

where $occ^{\sigma}_{\pi}(s, a)$ denotes the expected total number of occurrences of (s, a) under σ and π.

Proof. If (s, a) is not contained in a SEC of $\mathcal{M}$, then starting from s and under action a, the probability of returning to s is less than one, independent of the choice of strategy and nature. The proof then follows from basic results of probability theory. □

Proposition 15. *Let $E_{\mathcal{M}} = (S', \mathcal{A}')$ denote a SEC of IMDP $\mathcal{M}$. Then, we have $\sup_{\sigma \in \Sigma}\{ ExpTot^{\sigma,\infty}_{\mathcal{M}}[\mathsf{r}] \mid \mathcal{M}\!\downharpoonright_{\sigma} \models_{\Pi} ([T_1]^{\le k_1}_{\sim p_1}, \dots, [T_n]^{\le k_n}_{\sim p_n}) \} = \infty$ for a reward structure r of $\mathcal{M}$ if and only if there exists a strategy σ of $\mathcal{M}$ that $\mathcal{M}\!\downharpoonright_{\sigma} \models_{\Pi} ([T_1]^{\le k_1}_{\sim p_1}, \dots, [T_n]^{\le k_n}_{\sim p_n})$, $E_{\mathcal{M}}$ is reachable under σ, and $\mathsf{r}(\xi[i], \xi(i)) > 0$, where ξ is a path under σ with $\xi[i] \in S'$ and $(\xi[i], \xi(i)) \in \mathcal{A}'(\xi[i])$ for some $i \ge 0$.*

We can now construct, from $\mathcal{M}$, an *IMDP* $\bar{\mathcal{M}}$ that is equivalent to $\mathcal{M}$ in terms of satisfaction of φ but does not include actions with positive rewards in its SEC. The algorithm is similar to the one introduced in [Forejt et al. 2011] for *MDP*s and is as follows. First, remove action a from $\mathcal{A}(s)$ if (s, a) is contained in a SEC and $\mathsf{r}(s, a) > 0$ for some maximising reward structure r. Second, recursively remove states with no outgoing transitions and transitions that lead to non-existent states until a fixed point is reached.

Corollary 16. *There is a strategy σ of $\mathcal{M}$ such that $ExpTot^{\sigma,\infty}_{\mathcal{M}}[\mathsf{r}] = x < \infty$ and $\mathcal{M}\!\downharpoonright_{\sigma} \models_{\Pi} \varphi$ if and only if there is a strategy $\bar{\sigma}$ of $\bar{\mathcal{M}}$ such that $ExpTot^{\bar{\sigma},\infty}_{\bar{\mathcal{M}}}[\mathsf{r}] = x$ and $\bar{\mathcal{M}}\!\downharpoonright_{\bar{\sigma}} \models_{\Pi} \varphi$.*

3.3 Multi-Objective Robust Strategy Synthesis

We first study the computational complexity of multi-objective robust strategy synthesis problem for *IMDP*s. Formally,

Theorem 17. *Given an IMDP $\mathcal{M}$ and a multi-objective predicate φ, the problem of synthesising a strategy $\sigma \in \Sigma$ such that $\mathcal{M}\!\downharpoonright_{\sigma} \models_{\Pi} \varphi$ is **PSPACE**-hard.*

As the first step towards derivation of a solution approach for the robust strategy synthesis problem, we need to convert all reachability predicates to reward predicates and therefore, to transform an arbitrarily given query to a query over a basic predicate on a modified *IMDP*. This can be achieved simply by adding a reward of one at the time of reaching the target set and also negating the objective of predicates with upper-bounded relational operators. We correct and extend the procedure proposed in [Forejt et al. 2012] to reduce a general multi-objective predicate on an *IMDP* model to a basic form on a modified *IMDP*.

Proposition 18. *Given an IMDP* $\mathcal{M} = (S, \bar{s}, \mathcal{A}, I)$ *and a multi-objective predicate* $\varphi = ([T_1]^{\leq k_1}_{\sim_1 p_1}, \dots, [T_n]^{\leq k_n}_{\sim_n p_n}, [\mathsf{r}_{n+1}]^{\leq k_{n+1}}_{\sim_{n+1} r_{n+1}}, \dots, [\mathsf{r}_m]^{\leq k_m}_{\sim_m r_m})$*, let* $\mathcal{M}' = (S', \bar{s}', \mathcal{A}', I')$ *be the IMDP whose components are defined as follows:*

- $S' = S \times 2^{\{1,\dots,n\}}$;
- $\bar{s}' = (\bar{s}, \emptyset)$;
- $\mathcal{A}' = \mathcal{A} \times 2^{\{1,\dots,n\}}$; *and*
- *for all* $s, s' \in S$, $a \in \mathcal{A}$, *and* $v, v', v'' \subseteq \{1, \dots, n\}$,

$$I'((s,v),(a,v'),(s',v'')) = \begin{cases} I(s,a,s') & \text{if } v' = \{\, i \mid s \in T_i \,\} \setminus v \text{ and } v'' = v \cup v', \\ 0 & \text{otherwise.} \end{cases}$$

Now, let $\varphi' = ([\mathsf{r}_{T_1}]^{\leq k_1+1}_{\geq p'_1}, \dots, [\mathsf{r}_{T_n}]^{\leq k_n+1}_{\geq p'_n}, [\bar{\mathsf{r}}_{n+1}]^{\leq k_{n+1}}_{\geq r'_{n+1}}, \dots, [\bar{\mathsf{r}}_m]^{\leq k_m}_{\geq r'_m})$ *where, for each* $i \in \{1, \dots, n\}$,

$$p'_i = \begin{cases} p_i & \text{if } \sim_i = \geq, \\ -p_i & \text{if } \sim_i = \leq; \end{cases} \qquad \text{and} \qquad \mathsf{r}_{T_i}((s,v),(a,v')) = \begin{cases} 1 & \text{if } i \in v' \text{ and } \sim_i = \geq, \\ -1 & \text{if } i \in v' \text{ and } \sim_i = \leq, \\ 0 & \text{otherwise;} \end{cases}$$

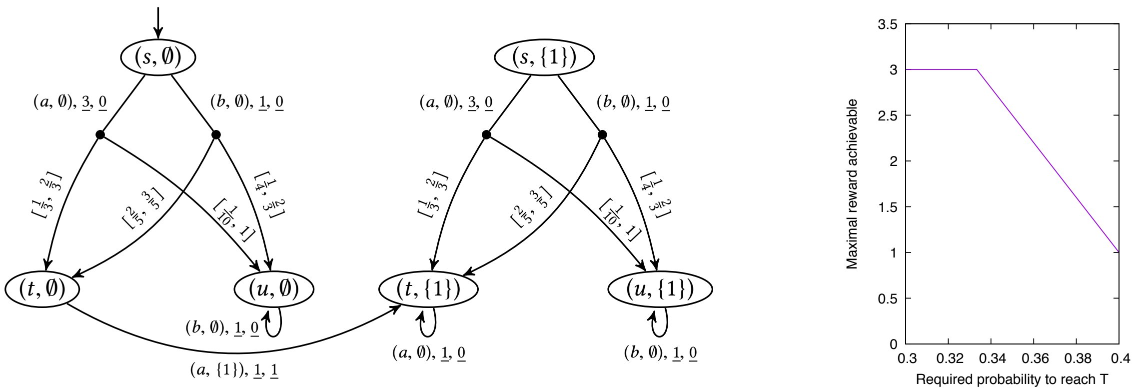
and, for each $j \in \{n+1, \dots, m\}$,

$$r'_j = \begin{cases} r_j & \text{if } \sim_j = \geq, \\ -r_j & \text{if } \sim_j = \leq; \end{cases} \qquad \text{and} \qquad \bar{\mathsf{r}}_j((s,v),(a,v')) = \begin{cases} \mathsf{r}_j(s,a) & \text{if } \sim_j = \geq, \\ -\mathsf{r}_j(s,a) & \text{if } \sim_j = \leq. \end{cases}$$

Then φ *is satisfiable in* $\mathcal{M}$ *if and only if* φ' *is satisfiable in* $\mathcal{M}'$.

Intuitively, the transformation of $\mathcal{M}$ to $\mathcal{M}'$ works as follows: for the reachability predicates, we transform them to reward predicates by assigning a reward of 1 the first time a state in the target set is reached; the information about which target sets have been reached is kept in the $v \subseteq \{1, \dots, n\}$ component of the transformed state. For both the original and the newly added reward predicates, we just transform the minimisation of positive rewards to the maximisation of their negative values, so all rewards are maximised. By doing this, we also make the threshold in the predicate comparison negative, e.g., we transform $[T_i]^{\leq k_i}_{\leq p_i}$ to $[\mathsf{r}_{T_i}]^{\leq k_i+1}_{\geq -p_i}$ and $[\mathsf{r}_j]^{\leq k_j}_{\leq r_j}$ to $[-\mathsf{r}_j]^{\leq k_j}_{\geq -r_j}$.

In [Forejt et al. 2012, Proposition 2] the thresholds are not made negative, and this is a flaw: consider for instance the *IMDP* $\mathcal{M}$ which has only two states, the initial s_0 and s_1, and the non-$[0,0]$ transitions $I(s_0, a, s_0) = I(s_0, b, s_1) = [1, 1]$; let $\varphi = ([\{s_1\}]^{\leq 1}_{\leq 0.5})$. Clearly $\mathcal{M}\!\downharpoonright_\sigma \models_\Pi \varphi$, by σ being the strategy choosing a in s_0. In the transformed *IMDP* $\mathcal{M}'$, the newly added reward structure $\mathsf{r}_{\{s_1\}}$ assigns reward 0 to $((s_0, \emptyset), (a, \emptyset))$ and reward -1 to $((s_0, \emptyset), (b, \{1\}))$; φ is transformed to $\varphi' = [\mathsf{r}_{\{s_1\}}]^{\leq 2}_{\geq -0.5}$, which is still satisfiable by the strategy choosing $(a, \emptyset)$ in $(s_0, \emptyset)$. Since $\mathcal{M}$ is also an *MDP*, we can apply the transformation given in [Forejt et al. 2012, Proposition 2]: $\mathcal{M}'$ and $\mathsf{r}_{\{s_1\}}$ are the same while φ is transformed to $\psi = [\mathsf{r}_{\{s_1\}}]^{\leq 2}_{\geq 0.5}$ (instead of $[\mathsf{r}_{\{s_1\}}]^{\leq 2}_{\geq -0.5}$), which is obviously unsatisfiable given that $\mathsf{r}_{\{s_1\}}$ assigns only non-positive values to each state-action pair.



(a) The transformed *IMDP* $\mathcal{M}'$

(b) Pareto curve.

Fig. 2. Example of *IMDP* transformation. (a) The *IMDP* $\mathcal{M}'$ generated from $\mathcal{M}$ shown in Fig. 1. (b) Pareto curve for the property $([\mathsf{r}_T]^{\leq 2}_{\max}, [\mathsf{r}]^{\leq 1}_{\max})$.

Example 19. To illustrate the transformation presented in Proposition 18, consider again the *IMDP* depicted in Fig. 1. Assume that the target set is $T = \{t\}$ and consider the property $\varphi = ([T]^{\leq 1}_{\geq \frac{1}{3}}, [\mathsf{r}]^{\leq 1}_{\geq \frac{1}{4}})$. The reduction converts φ to the property $\varphi' = ([\mathsf{r}_T]^{\leq 2}_{\geq \frac{1}{3}}, [\mathsf{r}]^{\leq 1}_{\geq \frac{1}{4}})$ on the modified $\mathcal{M}'$ depicted in Fig. 2a. We show two different reward structures $\bar{\mathsf{r}}$ and r_T besides each action, respectively.

In Fig. 2b we show the Pareto curve for this property. As we see, the maximal reward value is 3 as long as we require a probability at most $\frac{1}{3}$ to reach T. Afterwards, the reward obtainable linearly decreases. If we require a reachability probability for T of $\frac{2}{5}$, the reward obtained is just 1. For higher required probabilities and rewards, the problem becomes infeasible. The reason for this behaviour is that, as long as we do not require the reachability probability for T to be higher than $\frac{1}{3}$, action a can be chosen in state s, because the lower interval bound to reach t is $\frac{1}{3}$, which in turn leads to a reward of 3 being obtained. For higher reachability probabilities required, choosing action b with a certain probability is required, which however provides a lower reward. There is no strategy with which t is reached with a probability larger than $\frac{2}{5}$. ♢

By means of Proposition 18, for robust strategy synthesis we therefore need to only consider the basic multi-objective predicates of the form $([\mathsf{r}_1]^{\leq k_1}_{\geq r_1}, \dots, [\mathsf{r}_n]^{\leq k_n}_{\geq r_n})$. For such a predicate, we define its Pareto curve as follows.

Definition 20 (Pareto Curve of a Multi-objective Predicate). Given an *IMDP* $\mathcal{M}$ and a basic multi-objective predicate $\varphi = ([\mathsf{r}_1]^{\leq k_1}_{\geq r_1}, \dots, [\mathsf{r}_n]^{\leq k_n}_{\geq r_n})$, we define the set of achievable values with respect to φ as $A_{\mathcal{M},\varphi} = \{\, (r_1, \dots, r_n) \in \mathbb{R}^n \mid ([\mathsf{r}_1]^{\leq k_1}_{\geq r_1}, \dots, [\mathsf{r}_n]^{\leq k_n}_{\geq r_n})$ is satisfiable $\}$. We define the Pareto curve of φ, denoted $\mathcal{P}_{\mathcal{M},\varphi}$, to be the Pareto curve of $A_{\mathcal{M},\varphi}$.

It is not difficult to see that the Pareto curve is in general an infinite set, and therefore, it is usually not possible to derive an exact representation of it in polynomial time. However, it can be shown that an ε-approximation of it can be computed efficiently [Etessami et al. 2007].

In the remainder of this section, we describe an algorithm to solve the synthesis query. We follow the well-known *normalisation* approach in order to solve the multi-objective predicate which is essentially based on normalising multiple

Algorithm 1: Algorithm for solving robust synthesis queries

Input: An *IMDP* $\mathcal{M}$, multi-objective predicate $\varphi = ([\mathsf{r}_1]^{\leq k_1}_{\geq r_1}, \dots, [\mathsf{r}_n]^{\leq k_n}_{\geq r_n})$
Output: true if there exists a strategy $\sigma \in \Sigma$ such that $\mathcal{M}{\downharpoonright_{\sigma}} \models_{\Pi} \varphi$, false if not.

1 **begin**
2 $\quad X := \emptyset$;
3 $\quad \mathsf{r} := (\mathsf{r}_1, \dots, \mathsf{r}_n)$;
4 $\quad \mathbf{k} := (k_1, \dots, k_n)$;
5 $\quad \mathbf{r} := (r_1, \dots, r_n)$;
6 $\quad$ **while** $\mathbf{r} \notin X{\downarrow}$ **do**
7 $\quad\quad$ Find $\mathbf{w}$ separating $\mathbf{r}$ from $X{\downarrow}$;
8 $\quad\quad$ Find strategy σ maximising $\mathit{ExpTot}^{\sigma,\mathbf{k}}_{\mathcal{M}}[\mathbf{w} \cdot \mathsf{r}]$;
9 $\quad\quad$ $\mathbf{g} := (\mathit{ExpTot}^{\sigma,k_i}_{\mathcal{M}}[\mathsf{r}_i])_{1 \leq i \leq n}$;
10 $\quad\quad$ **if** $\mathbf{w} \cdot \mathbf{g} < \mathbf{w} \cdot \mathbf{r}$ **then**
11 $\quad\quad\quad$ **return** false;
12 $\quad\quad$ $X := X \cup \{\mathbf{g}\}$;
13 $\quad$ **return** true;

objectives into one single objective. It is known that the optimal solution of the normalised (single-objective) predicate, if it exists, is the Pareto optimal solution of the multi-objective predicate [Ehrgott 2006].

The robust synthesis procedure is detailed in Algorithm 1. This algorithm aims to construct a sequential approximation to the Pareto curve $\mathcal{P}_{\mathcal{M},\varphi}$ while the quality of approximations gets better and more precise with each iteration. In other words, along the course of Algorithm 1 a sequence of weight vectors $\mathbf{w}$ are generated and corresponding to each of them, a $\mathbf{w}$-weighted sum of n objectives is optimised through lines 8-9. The optimal strategy σ is then used in order to generate a point $\mathbf{g}$ on the Pareto curve $\mathcal{P}_{\mathcal{M},\varphi}$. We collect all these points in the set X. The multi-objective predicate φ is satisfiable once we realise that $\mathbf{r}$ belongs to $X{\downarrow}$.

The optimal strategies for the multi-objective robust synthesis queries are constructed following the approach of [Forejt et al. 2012] and as a result of termination of Algorithm 1. In particular, when Algorithm 1 terminates, a sequence of points $\mathbf{g}^1, \dots, \mathbf{g}^t$ on the Pareto curve $\mathcal{P}_{\mathcal{M},\varphi}$ are generated each of which corresponds to a deterministic strategy $\sigma_{\mathbf{g}^j}$ for the current point $\mathbf{g}^j$. The resulting optimal strategy σ_{opt} is subsequently constructed from these using a randomised weight vector $\alpha \in \mathbb{R}^t$ satisfying $r_i \leq \sum_{j=1}^{t} \alpha_i \cdot {g_i}^j$, as we will explain in Section 4.

Remark 21. *It is worthwhile to mention that the synthesis query for IMDPs cannot be solved on the MDPs generated from IMDPs by computing all feasible extreme transition probabilities and then applying the algorithm of [Forejt et al. 2012]. The latter is a valid approach provided the cooperative semantics is applied for resolving the two sources of nondeterminism in IMDPs. With respect to the competitive semantics needed here, one can instead transform IMDPs to $2\frac{1}{2}$-player games [Basset et al. 2014] and then along the lines of the previous approach apply the algorithm of [Chen et al. 2013a]. Unfortunately, the transformation to (MDPs or) $2\frac{1}{2}$-player games induces an exponential blowup, adding an exponential factor to the worst case time complexity of the decision problem. Our algorithm avoids this by solving the robust synthesis problem directly on the IMDP so that the core part, i.e., lines 8-9 of Algorithm 1 can be solved with time complexity polynomial in $|\mathcal{M}|$.*

Algorithm 2 represents a value iteration-based algorithm which extends the value iteration-based algorithm of [Forejt et al. 2012] and adjusts it for *IMDP* models by encoding the notion of robustness. More precisely, the core difference is at lines 7 and 19, where the optimal strategy is computed so as to be robust against any choice of nature.

Theorem 22. *Algorithm 1 is sound, complete, and has runtime exponential in $|\mathcal{M}|$, $\mathbf{k}$, and n.*

Algorithm 2: Value iteration-based algorithm to solve lines 6-7 of Algorithm 1

Input: An *IMDP* $\mathcal{M}$, weight vector w, reward structures $\mathsf{r} = (\mathsf{r}_1, \ldots, \mathsf{r}_n)$, time-bound vector $\mathbf{k} \in (\mathbb{N} \cup \{\infty\})^n$, threshold ε

Output: strategy σ maximising $ExpTot_{\mathcal{M}}^{\sigma,\mathbf{k}}[\mathbf{w} \cdot \mathsf{r}]$, $\mathrm{g} := (ExpTot_{\mathcal{M}}^{\sigma,k_i}[\mathsf{r}_i])_{1 \le i \le n}$

begin
- $\mathbf{x} := 0; \mathbf{x}^1 := 0; \ldots; \mathbf{x}^n := 0;$
- $\mathbf{y} := 0; \mathbf{y}^1 := 0; \ldots; \mathbf{y}^n := 0;$
- $\sigma^\infty(s) := \perp$ for all $s \in S$;
- **while** $\delta > \varepsilon$ **do**
  - **foreach** $s \in S$ **do**
    - $y_s := \max_{a \in \mathcal{A}(s)} (\sum_{\{ i | k_i = \infty \}} w_i \cdot \mathsf{r}_i(s, a) + \min_{\mathfrak{h}_s^a \in \mathcal{H}_s^a} \sum_{s' \in S} \mathfrak{h}_s^a(s') \cdot x_{s'});$
    - $\sigma^\infty(s) := \arg\max_{a \in \mathcal{A}(s)} (\sum_{\{ i | k_i = \infty \}} w_i \cdot \mathsf{r}_i(s, a) + \min_{\mathfrak{h}_s^a \in \mathcal{H}_s^a} \sum_{s' \in S} \mathfrak{h}_s^a(s') \cdot x_{s'});$
    - $\bar{\mathfrak{h}}_s^{\sigma^\infty(s)}(s') := \arg\min_{\mathfrak{h}_s^a \in \mathcal{H}_s^a} \sum_{s' \in S} \mathfrak{h}_s^a(s') \cdot x_{s'};$
  - $\delta := \max_{s \in S}(y_s - x_s);$
  - $\mathbf{x} := \mathbf{y};$
- **while** $\delta > \varepsilon$ **do**
  - **foreach** $s \in S$ *and* $i \in \{1, \ldots, n\}$ **where** $k_i = \infty$ **do**
    - $y_s^i := \mathsf{r}_i(s, \sigma^\infty(s)) + \sum_{s' \in S} \bar{\mathfrak{h}}_s^{\sigma^\infty(s)}(s') \cdot x_{s'}^i;$
  - $\delta := \max_{i=1}^n \max_{s \in S}(y_s^i - x_s^i);$
  - $\mathbf{x}^1 := \mathbf{y}^1; \ldots; \mathbf{x}^n := \mathbf{y}^n;$
- **for** $j = \max\{ k_b < \infty \mid b \in \{1, \ldots, n\} \}$ **down to** 1 **do**
  - **foreach** $s \in S$ **do**
    - $y_s := \max_{a \in \mathcal{A}(s)} (\sum_{\{ i | k_i \ge j \}} w_i \cdot \mathsf{r}_i(s, a) + \min_{\mathfrak{h}_s^a \in \mathcal{H}_s^a} \sum_{s' \in S} \mathfrak{h}_s^a(s') \cdot x_{s'});$
    - $\sigma^j(s) := \arg\max_{a \in \mathcal{A}(s)} (\sum_{\{ i | k_i \ge j \}} w_i \cdot \mathsf{r}_i(s, a) + \min_{\mathfrak{h}_s^a \in \mathcal{H}_s^a} \sum_{s' \in S} \mathfrak{h}_s^a(s') \cdot x_{s'});$
    - $\bar{\mathfrak{h}}_s^{\sigma^j(s)}(s') := \arg\min_{\mathfrak{h}_s^a \in \mathcal{H}_s^a} \sum_{s' \in S} \mathfrak{h}_s^a(s') \cdot x_{s'};$
    - **foreach** $i \in \{1, \ldots, n\}$ **where** $k_i \ge j$ **do**
      - $y_s^i := \mathsf{r}_i(s, \sigma^j(s)) + \sum_{s' \in S} \bar{\mathfrak{h}}_s^{\sigma^j(s)}(s') \cdot x_{s'}^i;$
    - $\mathbf{x} := \mathbf{y}; \mathbf{x}^1 := \mathbf{y}^1; \ldots; \mathbf{x}^n := \mathbf{y}^n;$
- **for** $i = 1$ **to** n **do**
  - $g_i := y_{\bar{s}}^i;$
- σ acts as σ^j in j^{th} step when $j < \max_{i \in \{1,\ldots,n\}} k_i$ and as σ^∞ afterwards;
- **return** σ, g;

Remark 23. *It is worthwhile to mention that our robust strategy synthesis approach can also be applied to MDPs with richer formalisms for uncertainties such as likelihood or ellipsoidal uncertainties while preserving the computational complexity. In particular, in every inner optimisation problem in Algorithm 1, the optimality of a Markovian deterministic strategy and nature is guaranteed as long as the uncertainty set is convex, the set of actions is finite and the inner optimisation problem which minimises/maximises the objective function over the choices of nature achieves its optimum (cf. [Puggelli 2014, Proposition 4.1]). Furthermore, due to the convexity of the generated optimisation problems, the computational complexity of our approach remains intact.*

3.4 Multi-Objective Quantitative Queries

In this section we discuss multi-objective quantitative queries and present algorithms to solve them. In particular, we follow the same direction as [Forejt et al. 2012] and show how Algorithm 1 can be adapted to solve these types of queries.

Algorithm 3: Algorithm for solving robust quantitative queries

```
Input: An IMDP 𝓜, objective [r_1]_max^{≤k_1}, multi-objective predicate ([r_2]_{≥r_2}^{≤k_2}, ..., [r_n]_{≥r_n}^{≤k_n})
Output: value of qnt([r_1]_max^{≤k_1}, ([r_2]_{≥r_2}^{≤k_2}, ..., [r_n]_{≥r_n}^{≤k_n}))
1  begin
2      X = ∅;
3      r = (r_1, ..., r_n);
4      k = (k_1, ..., k_n);
5      r = (min_{σ∈Σ} ExpTot_𝓜^{σ,k}[r_1], r_2, ..., r_n);
6      while r ∉ X↓ or w · g > w · r do
7          Find w separating r from X↓ such that w_1 > 0;
8          Find strategy σ maximising ExpTot_𝓜^{σ,k}[w · r];
9          g := (ExpTot_𝓜^{σ,k_i}[r_i])_{1≤i≤n};
10         if w · g < w · r then
11             return ⊥;
12         X = X ∪ {g};
13         r_1 := max{r_1, max{ r' | (r', r_2, ..., r_n) ∈ X↓ }};
14     return r_1;
```

To present the algorithm, consider the quantitative query $qnt([\mathsf{r}_1]_{\max}^{\leq k_1}, ([\mathsf{r}_2]_{\geq r_2}^{\leq k_2}, \dots, [\mathsf{r}_n]_{\geq r_n}^{\leq k_n})$. Algorithm 3, similarly to Algorithm 1, generates a sequence of points $\mathbf{g}$ on the Pareto curve from a sequence of weight vectors $\mathbf{w}$. In order to optimise the objective r_1, a sequence of lower bounds r_1 is generated which are used in the same manner as Algorithm 1. In particular, in the initial step we let r_1 be the minimum value for r_1 that can be computed with an instance of value iteration [Puggelli 2014]. The sequence of non-decreasing values for r_1 are generated at the next steps based on the set of points X specified so far. In each step, the computation in the lines 8-9 of Algorithm 3 can again be achieved using Algorithm 2.

At this point it is worthwhile to mention that Algorithm 3 is different from its counterpart [Forejt et al. 2012, Algorithm 3] especially concerning lines 5, 8-9. In fact, all computations in these lines are performed while considering the behaviour of an adversarial nature as detailed in Algorithm 2.

3.5 Multi-Objective Pareto Queries

We finally provide an algorithmic solution to compute Pareto queries. As for Algorithm 3, this algorithm is in fact designed as an adaption of Algorithm 1 as detailed below.

Our algorithm to solve Pareto queries is depicted as Algorithm 4 which is in principle an extension of its counterpart for *MDP*s [Forejt et al. 2012, Algorithm 4]. Similarly to Algorithm 3, the key differences of this algorithm with its counterpart are in lines 5-6 and 11-12. We present the algorithm with respect to two objectives; note that it can be extended easily to any finite number of objectives. Since the number of faces of the Pareto curve is exponentially large in the size of the model, the step bound, and the number of objectives and also the result of the value iteration algorithm to compute the individual points is an approximation, Algorithm 4 only constructs an ε-approximation of the Pareto curve.

Algorithm 4: Algorithm for solving robust Pareto queries

Input: An *IMDP* $\mathcal{M}$, reward structures $\mathsf{r} = (\mathsf{r}_1, \mathsf{r}_2)$, time bounds (k_1, k_2), $\varepsilon \in \mathbb{R}_{\geq 0}$
Output: An ε-approximation of the Pareto curve

begin
  $X = \emptyset$;
  $Y : \mathbb{R}^2 \to 2^{\mathbb{R}^2}$ with initial $Y(x) = \emptyset$ for all x;
  $\mathbf{w} = (1, 0)$;
  Find strategy σ maximising $ExpTot^{\sigma,\mathbf{k}}_{\mathcal{M}}[\mathbf{w} \cdot \mathsf{r}]$;
  $\mathbf{g} := (ExpTot^{\sigma,k_1}_{\mathcal{M}}[\mathsf{r}_1], ExpTot^{\sigma,k_2}_{\mathcal{M}}[\mathsf{r}_2])$;
  $X := X \cup \{\mathbf{g}\}$;
  $Y(\mathbf{g}) := Y(\mathbf{g}) \cup \{\mathbf{w}\}$;
  $\mathbf{w} := (0, 1)$;
  while $\mathbf{w} \neq \bot$ **do**
    Find strategy σ maximising $ExpTot^{\sigma,\mathbf{k}}_{\mathcal{M}}[\mathbf{w} \cdot \mathsf{r}]$;
    $\mathbf{g} := (ExpTot^{\sigma,k_1}_{\mathcal{M}}[\mathsf{r}_1], ExpTot^{\sigma,k_2}_{\mathcal{M}}[\mathsf{r}_2])$;
    $X := X \cup \{\mathbf{g}\}$;
    $Y(\mathbf{g}) := Y(\mathbf{g}) \cup \{\mathbf{w}\}$;
    $\mathbf{w} := \bot$;
    Order X to a sequence $\mathbf{x}^1, \dots, \mathbf{x}^m$ such that $\forall i : x^i_1 \leq x^{i+1}_1$ and $x^i_2 \geq x^{i+1}_2$;
    for $i = 1$ **to** m **do**
      Let $\mathbf{u}$ be the element of $Y(\mathbf{x}^i)$ with maximal u_1;
      Let $\mathbf{u}'$ be the element of $Y(\mathbf{x}^{i+1})$ with minimal u'_1;
      Find a point $\mathbf{p}$ such that $\mathbf{u} \cdot \mathbf{p} = \mathbf{u} \cdot \mathbf{x}^i$ and $\mathbf{u}' \cdot \mathbf{p} = \mathbf{u}' \cdot \mathbf{x}^{i+1}$;
      if *distance of* $\mathbf{p}$ *from* $X{\downarrow}$ *is* $\geq \varepsilon$ **then**
        Find $\mathbf{w}$ separating $X{\downarrow}$ from $\mathbf{p}$, maximising $\mathbf{w} \cdot \mathbf{p} - \max_{\mathbf{x} \in X{\downarrow}} \mathbf{w} \cdot \mathbf{x}$;
        break;
  return X;

3.6 PLTL and ω-regular Properties

PLTL formulas, or in general ω-regular properties, allow one to express properties of an *IMDP* with respect to its infinite behaviour. Examples of PLTL formulas are: with probability at least 0.95, the *IMDP* will never be trapped in an error state ($Pr_{\geq 0.95}[\mathbf{GF}\neg error]$); almost surely, whenever a request arrives, eventually a response is provided ($Pr_{\geq 1}[\mathbf{G}(req \implies \mathbf{F}resp)]$); with probability at least 0.99, the system eventually becomes stable ($Pr_{\geq 0.99}[\mathbf{FG}stable]$). The classical approach to verify a PLTL formula $Pr_{\bowtie p}[\Psi]$, or an ω-regular property, against an *MDP* M consists in constructing a deterministic Rabin automaton (DRA) $\mathcal{R}_\Psi$ accepting the same words satisfying Ψ, then construct the product $\mathsf{M} \times \mathcal{R}_\Psi$, find the accepting maximal end components of $\mathsf{M} \times \mathcal{R}_\Psi$, and then compute the probability of reaching the union of such end components. We refer the interested reader to [Baier and Katoen 2008] for more details.

In the remaining part of this section we present how to analyse ω-regular properties against an *IMDP* $\mathcal{M}$. In practice, the construction is the extension to *IMDP*s of the approach for *MDP*s.

Definition 24 (Product IMDP $\mathcal{M} \times \mathcal{R}$). For given *IMDP* $\mathcal{M} = (S, \bar{s}, \mathcal{A}, I, AP, L)$ and DRA $\mathcal{R} = (Q, \bar{q}, 2^{AP}, T, Acc)$ with $Acc = \{(A_1, R_1), \dots, (A_k, R_k)\}$, the product $\mathcal{M} \times \mathcal{R}$ is the *IMDP* $\mathcal{M} \times \mathcal{R} = (S \times Q, \bar{s}', \mathcal{A}, I', Q, L')$ where

- $\bar{s}' = (\bar{s}, T(\bar{q}, L(\bar{s}))$;

- $I'((s,q),a,(s',q')) = \begin{cases} I(s,a,s') & \text{if } q' = T(q, L(s')), \\ [0,0] & \text{otherwise; and} \end{cases}$
- $L'(s,q) = \{q\}$.

Similarly to the *MDP* case, we can prove that the probability of $\mathcal{M}$ to satisfy Ψ equals the probability of reaching accepting SECs in $\mathcal{M} \times \mathcal{R}_\Psi$, where a SEC $\mathcal{M}'$ of $\mathcal{M} \times \mathcal{R}_\Psi$ with states S' and labelling L' is accepting if there exists $1 \leq i \leq k$ such that $A_i \cap L'(S') \neq \emptyset$ and $R_i \cap L'(S') = \emptyset$.

Theorem 25. *Let $\mathcal{M}$ be an IMDP, Ψ an LTL formula, and U be the union of all accepting SECs in $\mathcal{M} \times \mathcal{R}_\Psi$. Then for each strategy σ for $\mathcal{M}$ there exist a strategy σ' for $\mathcal{M} \times \mathcal{R}_\Psi$ such that for each nature π for $\mathcal{M}$ there exists a nature π' for $\mathcal{M} \times \mathcal{R}_\Psi$ such that*

$$\Pr^{\sigma,\pi}_{\mathcal{M}}[\{\, \xi \in \mathit{IPaths}_{\mathcal{M}} \mid \xi \models \Psi \,\}] = \Pr^{\sigma',\pi'}_{\mathcal{M}\times\mathcal{R}_\Psi}[\{\, \xi \in \mathit{IPaths}_{\mathcal{M}\times\mathcal{R}_\Psi} \mid \exists j \in \mathbb{N} : \xi[j] \in U \,\}]$$

and vice-versa.

Proof. The proof is a minor adaptation of the one for *MDP*s (cf. [Baier and Katoen 2008; Bianco and de Alfaro 1995]). Intuitively, strategy σ' is built out of σ as for the *MDP* setting, while nature π' is defined to mimic exactly π. □

As an immediate consequence of Theorem 25, we also have that the robust probability of satisfying Ψ under a strategy σ for $\mathcal{M}$ coincides with the robust probability of reaching accepting SECs under some strategy σ' for $\mathcal{M} \times \mathcal{R}_\Psi$.

Corollary 26. *Let $\mathcal{M}$ be an IMDP, $Pr_{\sim p}[\Psi]$ a PLTL formula, and U be the union of all accepting SECs in $\mathcal{M} \times \mathcal{R}_\Psi$; let Π' denote the set of natures for $\mathcal{M} \times \mathcal{R}_\Psi$. Then for each strategy σ for $\mathcal{M}$ there exists a strategy σ' for $\mathcal{M} \times \mathcal{R}_\Psi$ such that*

$$\operatorname*{opt}_{\pi \in \Pi} \Pr^{\sigma,\pi}_{\mathcal{M}}[\{\, \xi \in \mathit{IPaths} \mid \xi \models \Psi \,\}] = \operatorname*{opt}_{\pi' \in \Pi'} \Pr^{\sigma',\pi'}_{\mathcal{M}\times\mathcal{R}_\Psi}[\{\, \xi \in \mathit{IPaths} \mid \exists j \in \mathbb{N} : \xi[j] \in U \,\}]$$

and vice-versa, where $\mathrm{opt} = \min$ *if* $\sim = \geq$ *and* $\mathrm{opt} = \max$ *if* $\sim = \leq$.

By means of Theorem 25 and Corollary 26, we can extend the results about multi-objective (quantitative) queries (cf. Sec. 3.1 and 3.4) and Pareto queries (cf. Sec. 3.5) to general PLTL and ω-regular properties, by following a similar approach as shown in [Etessami et al. 2007].

4 GENERATION OF RANDOMISED STRATEGIES

In this section we describe how randomised strategies can be obtained as weighted sum of deterministic strategies. We consider a fixed *IMDP* $\mathcal{M} = (S, \bar{s}, \mathcal{A}, I)$ and a basic multi-objective predicate $([\mathtt{r}_1]^{\leq k_1}_{\geq r_1}, \dots, [\mathtt{r}_n]^{\leq k_n}_{\geq r_n})$. For clarity, we assume that all $k_i = \infty$; we discuss the extension to $k_i < \infty$ afterwards. In the following, we will describe how we can obtain a randomised strategy from the results computed by Algorithms 1, 3, and 4. These algorithms compute a set $X = \{\mathbf{g}_1, \dots, \mathbf{g}_m\}$ of reward vectors $\mathbf{g}_i = (g_{i,1}, \dots, g_{i,n})$ and their corresponding set of strategies $\Sigma = \{\sigma_1, \dots, \sigma_m\}$, where strategy σ_i achieves the reward vector $\mathbf{g}_i$.

In the descriptions of the given algorithms, the strategies σ_i are not explicitly stored and mapped to the reward they achieve, but they can be easily adapted. All used strategies are memoryless (due to the assumption that $k_i = \infty$) and deterministic; this means that we can treat them as functions of the form $\sigma_i : S \to \mathcal{A}$ or, equivalently, as functions $\sigma_i : S \times \mathcal{A} \to \{0,1\}$ where $\sigma_i(s,a) = 1$ if $\sigma_i(s) = a$ and $\sigma_i(s,\cdot) = 0$ otherwise.

From the set X, we can compute a set $P = \{p_1, \dots, p_m\}$ of the probabilities with which each of these strategies shall be executed. If we execute each σ_i with its according probability p_i, the vector of total expected rewards is $\mathbf{g} = \sum_{i=1}^{m} p_i \cdot \mathbf{g}_i$.

Let $\mathbf{r} = (r_1, \ldots, r_n)$ denote the vector of reward bounds of the multi-objective predicate. To obtain P after having executed Algorithm 1, we can choose the values p_i in P such that they fulfil the constraints $\sum_{i=1}^{m} \mathbf{g}_i \cdot p_i \geq \mathbf{r}$, $\sum_{i=1}^{m} p_i = 1$, and $p_i \geq 0$ for each $1 \leq i \leq m$. For the other algorithms, P can be computed accordingly.

To obtain a stochastic process with expected values $\mathbf{g}$, we initially randomly choose one of the memoryless deterministic strategies σ_i according to their probabilities in P. Afterwards, we just keep executing the chosen σ_i. The initial choice of the strategy to execute is the only randomised choice to be made. We do *not* perform a random choice after the initial choice of σ_i.

This process of obtaining the expected rewards $\mathbf{g}$ indeed uses memory, because we have to remember the deterministic strategy which was randomly chosen to be executed. On the other hand, we only need a very limited way of randomisation.

We like to emphasise that indeed we cannot just construct a memoryless randomised strategy by choosing the strategy σ_i with probability p_i in each step anew.

Example 27. Consider the *IMDP* in Fig. 3. We only have two possible actions, a and b. The initial state is s and all probability intervals are the interval $[1, 1]$, which we omit for readability; thus, there is also only one possible nature π. There is only a single reward structure, indicated by the underlined numbers. If we choose a in state s, we end up in t in the next step and obtain a reward of 1 with certainty, while if we choose b, we will be in u in the next step and obtain a reward of 0, and accordingly for the other states.

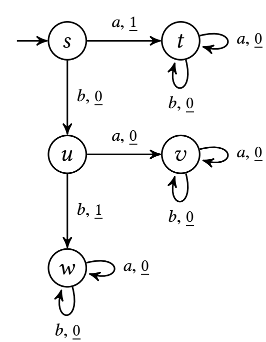


Fig. 3. Computing randomised strategies.

We consider the strategies σ_a which chooses a in each state and σ_b which chooses b in each state. With both strategies, we accumulate a reward of exactly 1. Therefore, if we choose to execute σ_a with probability 0.5 and σ_b with the same probability, this process will lead to a reward of 1 as well.

Now, consider a strategy which chooses the action selected by σ_a *in each* state with probability 0.5, and with the same probability chooses the action selected by σ_b. It is easy to see that this strategy only obtains a reward of $0.5 \cdot 1 + 0.5 \cdot 0.5 \cdot 1 = 0.75$. As we see, this naive way of combining the two deterministic strategies into a memoryless randomised strategy is not optimal. $\Diamond$

Thus, the way to construct a memoryless randomised strategy is somewhat more involved. We will have to compute the *state-action frequencies*, that is the average number of times a given state-action pair is seen.

At first, we fix an arbitrary memoryless nature $\pi\colon \mathit{FPaths} \times \mathcal{A} \to \mathrm{Disc}(S)$, that is, $\pi\colon S \times \mathcal{A} \to \mathrm{Disc}(S)$. The particular choice of π is not important, which is due to the fact that our algorithms are robust against any choice of nature. We then let $x_i^{\sigma}(s)$ denote the probability to be in state s at step i when strategy σ is used (using nature π and under the condition that we have started in $\bar{s}$).

For any $\sigma \in \Sigma$, we have $x_i^{\sigma}(s) = \sum_{\{\, \xi \in \mathit{FPaths} \mid \mathit{last}(\xi) = s, |\xi| = i \,\}} \mathrm{Pr}_{\mathcal{M}}^{\sigma,\pi}(\mathit{Cyl}_{\xi})$, which can be shown to be equivalent to the inductive form $x_0^{\sigma}(\bar{s}) = 1$ and $x_0^{\sigma}(s) = 0$ for $s \neq \bar{s}$, and $x_{i+1}^{\sigma}(s) = \sum_{s' \in S} \pi(s', \sigma(s'))(s) \cdot x_i^{\sigma}(s')$.

The state-action frequency $y^{\sigma}(s, a)$ is the number of times action a is chosen in state s when using strategy σ. We then have that $y^{\sigma}(s, a) = \sum_{i=0}^{\infty} x_i^{\sigma}(s) \cdot \sigma(s, a)$. Thus, state-action frequencies can be approximated using a simple value iteration scheme. The *mixed state-action frequency* $y(s, a)$ is the average over all state action frequencies weighted by the probability with which a given strategy is executed. Thus, $y(s, a) = \sum_{i=1}^{m} p_i \cdot y^{\sigma_i}(s, a)$ for all s, a. To construct a memoryless randomised strategy σ, we normalise the probabilities to $\sigma(s,a) = \frac{y(s,a)}{\sum_{b \in \mathcal{A}} y(s,b)}$ for all $s \in S$ and $a \in \mathcal{A}(s)$ (see also the description for the computation of strategies/adversaries below [Forejt et al. 2011, Proposition 4]).

Example 28. In the model of Fig. 3, we have $y^{\sigma_a}(s,a) = 1$, $y^{\sigma_a}(s,b) = 0$, $y^{\sigma_a}(u,a) = 0$, $y^{\sigma_a}(u,b) = 0$, $y^{\sigma_b}(s,a) = 0$, $y^{\sigma_b}(s,b) = 1$, $y^{\sigma_b}(u,a) = 0$, and $y^{\sigma_b}(u,b) = 1$. If we choose both σ_a and σ_b with probability 0.5, we obtain the mixed state-action frequencies $y(s,a) = 0.5$, $y(s,b) = 0.5$, $y(u,a) = 0$, and $y(u,b) = 0.5$. The memoryless randomised strategy σ we can construct is then $\sigma(s,a) = 0.5$, $\sigma(s,b) = 0.5$, $\sigma(u,a) = 0$, $\sigma(u,b) = 1$, which indeed achieves a reward of 1. $\Diamond$

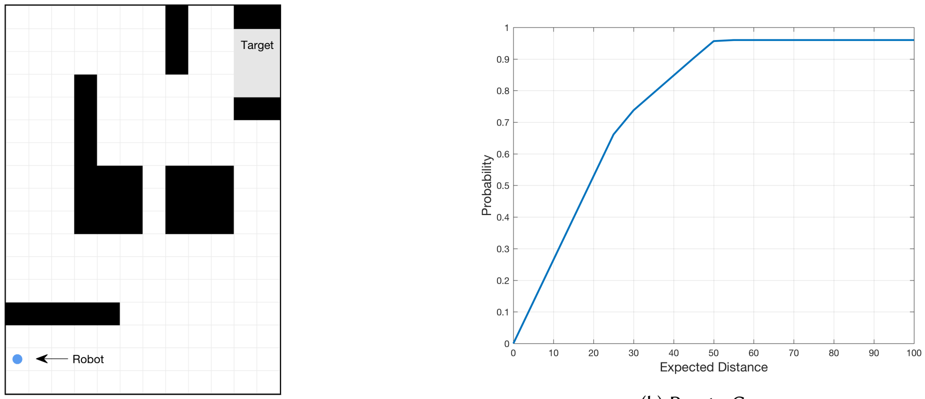
For the general case where $k_i < \infty$ for some k_i, we have to work with counting deterministic strategies and natures. Let $k_{\max}$ be the largest non-infinite step bound. The usage of memory is unavoidable here because it is required already in case of a single step-bounded objective. To achieve optimal values, the computed strategies have to be able to make their decision dependent on how many steps are left before the step bound is reached. Thus, we have strategies of the form $\sigma_i \colon S \times \{0, \dots, k_{\max}\} \to \mathcal{A}$ or equivalently $\sigma_i \colon S \times \{0, \dots, k_{\max}\} \times \mathcal{A} \to \{0,1\}$ where $\sigma_i(s,j,a) = 1$ if $\sigma_i(s,j) = a$ and $\sigma_i(s,j,\cdot) = 0$ otherwise. For step i with $i < k_{\max}$, a strategy σ chooses action $\sigma(s,i)$ for state s whereas for all $i \geq k_{\max}$ the decision $\sigma(s,k_{\max})$ is used. Natures are of the form $\pi \colon S \times \mathcal{A} \times \{0, \dots, k_{\max}\} \to \mathrm{Disc}(S)$. The computation of the randomised strategy changes accordingly: for any $\sigma \in \Sigma$, we have $x_0^{\sigma}(\bar{s}) = 1$, $x_0^{\sigma}(s) = 0$ for $s \neq \bar{s}$, and $x_{i+1}^{\sigma}(s) = \sum_{s' \in S} \pi(s', \sigma(s',i'), i')(s) \cdot x_{i'}^{\sigma}(s')$ where $i' = \min\{i, k_{\max}\}$. Also the state-action frequencies are now defined as step-dependent. For $i \in \{0, \dots, k_{\max} - 1\}$ we define $y^{\sigma}(s,i,a) = x_i^{\sigma}(s) \cdot \sigma(s,i,a)$ and $y^{\sigma}(s,k_{\max},a) = \sum_{i \geq k_{\max}} x_i^{\sigma}(s) \cdot \sigma(s,a)$.

The mixed state-action frequency is then $y(s,i,a) = \sum_{j=1}^{m} p_j \cdot y^{\sigma_j}(s,i,a)$. Again using normalisation we define the counting randomised strategy $\sigma(s,i,a) = \frac{y(s,i,a)}{\sum_{b \in \mathcal{A}} y(s,i,b)}$. Here, for step i with $i < k_{\max}$ we use decisions from $\sigma(\cdot, i, \cdot)$ while for $i \geq k_{\max}$ we use decisions from $\sigma(\cdot, k_{\max}, \cdot)$.

The bounded step case can be derived from the unbounded step case in the following sense: we can transform the *IMDP* and the predicate into an *unrolled IMDP*. Here, we encode the step bounds in the state space as follows: we copy the state space S a number of $k_{\max} + 1$ times to a new state space $S_{\mathrm{unrolled}} = \dot{\bigcup}_{i \in \{0, \dots, k_{\max}\}} S_i$. We call each set of states S_i a *layer*. For each state $s \in S$ and $i \in \{0, \dots, k_{\max}\}$ we have $s_i \in S_i$. If we have a transition from a state s to a state s', in the unrolled *IMDP* for all $i \in \{0, \dots, k_{\max} - 1\}$ we have an according transition from s_i to s_{i+1} instead. We also have a transition from $s_{k_{\max}}$ to $s'_{k_{\max}}$. Formally, for $i < k_{\max}$ we have $I^{\mathrm{unrolled}}(s_i, a, s'_{i+1}) = I(s,a,s')$ for some states s, s' and some action a and zero else, and then $I^{\mathrm{unrolled}}(s_{k_{\max}}, a, s'_{k_{\max}}) = I(s,a,s')$. Thus, there are only transitions from a one layer to the next layer, except for layer $k_{\max}$ which behaves like the original *IMDP*.

Reward structures are defined as follows. We assume that each reward property uses a different reward structure. For unbounded reward properties using reward structure r, we just let $\mathsf{r}^{\mathrm{unrolled}}(s_i, a) = \mathsf{r}(s,a)$ for all i and states s. For a step bounded reward property with bound k we define a modified reward structure as follows: for layers 0 to $k-1$, the reward is obtained as usual, that is $\mathsf{r}^{\mathrm{unrolled}}(s_i, a) = \mathsf{r}(s,a)$ for $i \in \{0, \dots, k-1\}$. However, to simulate the step bound, we let $\mathsf{r}(s_i, a) = 0$ for $i \geq k$.

By removing the step bound from predicate, we can now analyse the unrolled *IMDP* and obtain the same result as in the original *IMDP* using the original step bounded predicate. As we are considering only unbounded properties, we obtain a set of memoryless deterministic strategies. We can then construct a counting strategy for the original model by mapping the layer number to the step number, that is $\sigma(s,i,a) = \sigma^{\mathrm{unrolled}}(s_i, a)$. In this way, we can show the correctness of the above strategy computation for the step bounded case, because then also the values for the state action frequencies carry over, that is e.g. $y(s,i,a) = y^{\mathrm{unrolled}}(s_i, a)$. Note that for $i < k_{\max}$ in $y^{\mathrm{unrolled},\sigma}(s_i, a) = \sum_{j=0}^{\infty} x_j^{\sigma}(s_i) \cdot \sigma(s_i, a)$ only the summand for $j = i$ is relevant. This is the case because by construction of the unrolled *IMDP* for the other j with $j \neq i$



(a) Robot Environment

(b) Pareto Curve

Fig. 4. Simple-Task Robotic Scenario. (a) Environment map, where obstacles and target are shown in black and grey, respectively. (b) Pareto curve for the property $([\mathrm{r}_p]^{\leq\infty}_{\max}, [\mathrm{r}_d]^{\leq\infty}_{\min})$.

we have $x^\sigma_j(s_i) = 0$. Thus, $y^{\text{unrolled},\sigma}(s_i, a) = x^\sigma_i(s_i) \cdot \sigma(s_i, a)$. Accordingly, for $y^{\text{unrolled},\sigma}(s_{k_{\max}}, a) = \sum_{j=0}^{\infty} x^\sigma_j(s_{k_{\max}}) \cdot \sigma(s_{k_{\max}}, a)$ only j with $j \geq k_{\max}$ are relevant and thus $y^{\text{unrolled},\sigma}(s_{k_{\max}}, a) = \sum_{j\geq k_{\max}}^{\infty} x^\sigma_j(s_{k_{\max}}) \cdot \sigma(s_{k_{\max}}, a)$.

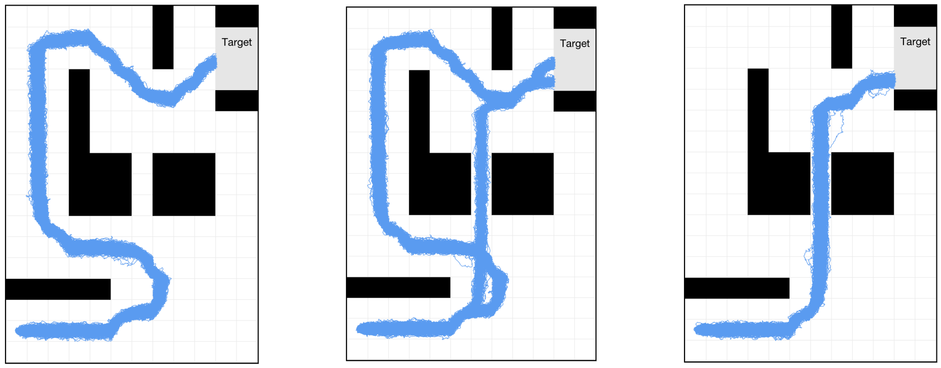
5 CASE STUDIES

We implemented the proposed multi-objective robust strategy synthesis algorithms and applied them to three case studies: (1) simple-task motion planning for a robot with noisy continuous dynamics, (2) motion planning for a warehouse robot with complex tasks, and (3) autonomous nondeterministic tour guides drawn from [Cantino et al. 2007; Hashemi et al. 2016]. All experiments took a few seconds to complete on a standard laptop PC.

5.1 Simple-Task Motion Planning under Uncertainty

In robot motion planning, designers often seek a plan that simultaneously satisfies multiple objectives [Lahijanian and Kwiatkowska 2016], e.g., *maximising the chances of reaching the target while minimising the energy consumption*. These objectives are usually in conflict with each other; hence, presenting the Pareto curve, i.e., the set of achievable points with optimal trade-off between the objectives, is helpful to the designers. They can then choose a point on the curve according to their desired guarantees and obtain the corresponding plan (strategy) for the robot. In this case study, we considered such a motion planning problem for a noisy robot with continuous dynamics in an environment with obstacles and a target region, as depicted in Fig. 4a. The robot's motion model was a single integrator with additive Gaussian noise. The initial state of the robot was on the bottom-left of the environment. The objectives were to reach the target safely while minimising the energy consumption, which is proportional to the travelled distance.

We approached this problem by first abstracting the motion of the noisy robot in the environment as an *IMDP* $\mathcal{M}$ and then computing strategies on $\mathcal{M}$ as in [Luna et al. 2014a,b,c]. The abstraction was achieved by partitioning the environment into a grid and computing local (continuous) controllers to allow transitions from every cell to each of its neighbours. The cells and the local controllers were then associated to the states and actions of the *IMDP*, respectively, resulting in 204 states (cells) and 4 actions per state. The boundaries of the environment were also associated with a



(a) φ_1 (b) φ_2 (c) φ_3

Fig. 5. Robot sample paths under strategies for φ_1, φ_2, and φ_3

state. Note that the transition probabilities between cells were raised by the noise in the dynamics and their ranges were due to variation of the possible initial robot (continuous) state within each cell.

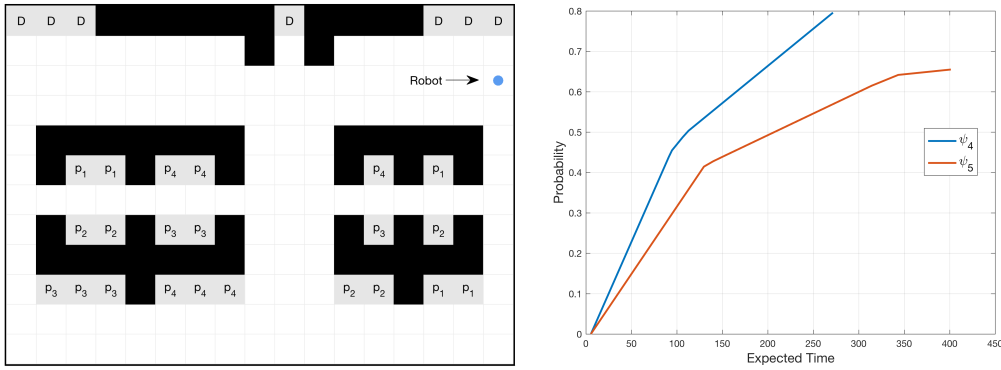
The guarantee that can be provided for the original continuous system is that the computed bounds (both for the probability of satisfaction and expected travelled distance) on the abstracted *IMDP* also hold for the continuous system (cf. [Luna et al. 2014b]). For a single robot, these bounds provide a measure of "goodness" of the robot's performance. For a swarm of robots, these bounds provide guarantees on the number of robots that can safety make it to the target while respecting the distance constraint.

The *IMDP* states corresponding to obstacles (including boundaries) were given deterministic self-transitions, modelling robot termination as the result of a collision. To allow for the computation of the probability of reaching target, we included an extra state in the *IMDP* with a deterministic self-transition and then added incoming deterministic transitions to this state from the target states. A reward structure r_p, which assigns a reward of 1 to these transitions and 0 to all the others, in fact, computes the probability of reaching the target. To capture the travelled distance, we defined a reward structure r_d assigning a reward of 0 to the state-action pairs with self-transitions and 1 to the rest.

The two robot objectives then can be expressed as: $([\mathsf{r}_p]^{\leq\infty}_{\max}, [\mathsf{r}_d]^{\leq\infty}_{\min})$. We first computed the Pareto curve for the property, which is shown in Fig. 4b, to find the set of all achievable values (optimal trade-offs) for the reachability probability and expected travelled distance. The Pareto curve shows that there is clearly a trade-off between the two objectives. To achieve high probability of reaching target safely, the robot needs to travel a longer distance, i.e., spend more energy, and vice versa. We chose three points on the curve and computed the corresponding robust strategies for

$$\varphi_1 = ([\mathsf{r}_p]^{\leq\infty}_{\geq 0.95}, [\mathsf{r}_d]^{\leq\infty}_{\leq 50}), \qquad \varphi_2 = ([\mathsf{r}_p]^{\leq\infty}_{\geq 0.90}, [\mathsf{r}_d]^{\leq\infty}_{\leq 45}), \qquad \varphi_3 = ([\mathsf{r}_p]^{\leq\infty}_{\geq 0.66}, [\mathsf{r}_d]^{\leq\infty}_{\leq 25}).$$

We then simulated the robot under each strategy 500 times. The statistical results of these simulations are consistent with the bounds in φ_1, φ_2, and φ_3. The collision-free robot trajectories are shown in Fig. 5. These trajectories illustrate that the robot is conservative under φ_1 and takes a longer route with open spaces around it to reach the target in order to be safe (Fig. 5a), while it becomes reckless under φ_3 and tries to go through a narrow passage with the knowledge that its motion is noisy and could collide with the obstacles (Fig. 5c). This risky behaviour, however, is required in order to meet the bound on the expected travelled distance in φ_3. The sample trajectories for φ_2 (Fig. 5b) demonstrate the



(a) Warehouse Environment

(b) Pareto Curves

Fig. 6. Warehouse Robotic Scenario. (a) Warehouse map, where the product pick-up locations and drop-off zones are shown in grey and obstacles in black. (b) Pareto curves for the properties $(Pr_{\max=?}[\psi_i], [\mathsf{r}_t]^{\leq\infty}_{\min})$ for $i \in \{4, 5\}$.

stochastic nature of the strategy. That is, the robot probabilistically chooses between being safe and reckless in order to satisfy the bounds in φ_2.

5.2 Warehouse Robot Planning with Complex Tasks

In this case study, we consider a warehouse scenario in which a robot is tasked to collect ordered products and deliver them to a drop-off zone. For optimal productivity, the robot should perform the tasks in the minimum amount of time and with the minimum amount of damages to itself and to the products by avoiding obstacles. The robot model is the same as the one in Sec. 5.1, and the warehouse map is shown in Fig. 6a. In this figure, the pick-up locations for product i is marked by p_i, and the drop-off zones are marked by D.

We constructed the *IMDP* model of this robot in the similar manner as in Sec. 5.1. We labelled the states of the *IMDP* with their propositions p_i for $1 \leq i \leq 4$, drop-off, and obstacle. Moreover, we assign a reward of 5 denoting the maximum duration of time (in seconds) it takes the robot to make a transition from one cell to another. The *IMDP* had a total of 205 states and 4 actions per state.

We consider two orders (tasks):

- "Pick up product p_1 and deliver it to a drop-off zone and always avoid obstacles," and
- "Pick up products p_1, p_2, and p_3 in any order and deliver them to a drop-off zone, and avoid drop-off zones until all three products are gathered, and always avoid obstacles."

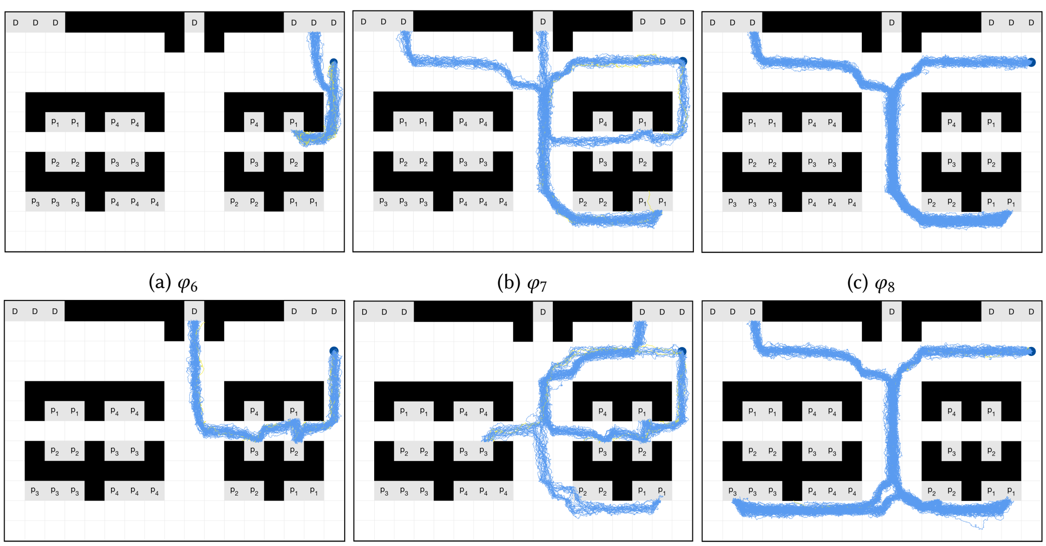
The corresponding LTL formulas, respectively, are:

$$\psi_4 = \mathbf{G}\neg\text{obstacle} \wedge \mathbf{F}(p_1 \wedge \mathbf{F}\text{drop-off}), \qquad \psi_5 = \mathbf{G}\neg\text{obstacle} \wedge \bigwedge_{i=1}^{3}(\neg\text{drop-off}\,\mathbf{U}\,p_i) \wedge \mathbf{F}\text{drop-off}.$$

Therefore, the pair of objectives for each task can be expressed as $(P_{\max}[\psi_i], [\mathsf{r}_t]^{\leq\infty}_{\min})$ for $i \in \{4, 5\}$, where r_t corresponds to the reward structure for time. To compute the Pareto curves, we first constructed the corresponding Rabin automata and the product *IMDP*s for tasks ψ_4 and ψ_5. The *IMDP*s had 617 and 2,462 states, respectively, and four actions per state. The Pareto curves for the above multi-objective formulas are shown in Fig. 6b. Then, we computed the robust strategies for the following properties (Pareto points):

$$\varphi_6 = (Pr_{\geq 0.43}[\psi_4], [\mathsf{r}_t]^{\leq\infty}_{\leq 90}), \qquad \varphi_7 = (Pr_{\geq 0.67}[\psi_4], [\mathsf{r}_t]^{\leq\infty}_{\leq 200}), \qquad \varphi_8 = (Pr_{\geq 0.80}[\psi_4], [\mathsf{r}_t]^{\leq\infty}_{\leq 270}),$$
$$\varphi_9 = (Pr_{\geq 0.41}[\psi_5], [\mathsf{r}_t]^{\leq\infty}_{\leq 130}), \qquad \varphi_{10} = (Pr_{\geq 0.49}[\psi_5], [\mathsf{r}_t]^{\leq\infty}_{\leq 200}), \qquad \varphi_{11} = (Pr_{\geq 0.65}[\psi_5], [\mathsf{r}_t]^{\leq\infty}_{\leq 400}).$$

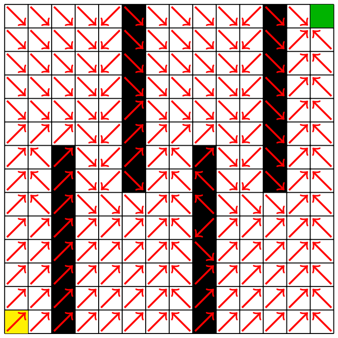
The sample robot trajectories under these strategies are shown in Fig. 7, where the initial position of the robot is indicated by a dark-blue disk. From the figures, it is evident that the robot chooses longer paths that are safer as more time is allowed. For properties φ_6-φ_8 that correspond to task ψ_4, the robot chooses the shortest path to p_1 by first going down through the narrow passage and then returning on the same path to the drop-off zone when only 90s are allowed (Fig. 7a). This path however has a higher risk to incur a damage. When 200s are given, the robot uses a mixture of two paths that are less risky as shown in Fig. 7b. One path leads the robot down, through the narrow passage, between the shelves, and finally straight up to the drop-off zone. The other path takes the robot left, then down through the middle of the warehouse to the bottom right p_1, returning on the similar path in the middle, and finally to the drop-off zone on the left side. For the bound of 270s, the robot chooses only the latter path, which is the safest path that has the most open spaces (Fig. 7c). A similar trend is observed for φ_9-φ_{11} but at larger time duration since task ψ_5 requires a collection of three products as shown in Figs. 7d-7f. Finally, we computed the probability and average time duration for 500 sample paths under each strategy, and the obtained values were within the bounds for φ_6-φ_{11}, validating the proposed approach.



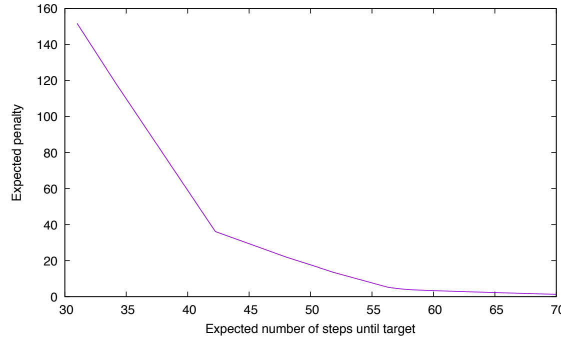
(a) φ_6 (b) φ_7 (c) φ_8

(d) φ_9 (e) φ_{10} (f) φ_{11}

Fig. 7. Robot sample paths under strategies for φ_6-φ_{11}. The robot's initial position is indicated by a dark-blue disk and the paths are: (a) *down-p_1-up-D*, (b) mixture of two paths of *down-p_1-middle-up-D* and *left-middle-down-p_1-middle-up-left-D*, (c) *left-middle-down-p_1-middle-up-left-D*, (d) *down-p_2-p_1-p_3-middle-up-D*, (e) mixture of two paths: *down-p_1-p_2-p_3-middle-up-right-D* and *left-middle-down-p_3-down-p_2-p_1-middle-up-right-D*, (f) *left-middle-down-right-p_1-p_2-p_3-middle-up-left-D*.



(a) The ANTG model for $n = 14$. The yellow, black, and green cells represent the entrance, closed, and exit parts of the museum, respectively. The red arrows indicate an example strategy.



(b) The Pareto Curve

Fig. 8. The ANTG case study: model and analysis

5.3 The Model of Autonomous Nondeterministic Tour Guides

Our second case study is inspired by "Autonomous Nondeterministic Tour Guides" (ANTG) in [Cantino et al. 2007; Hashemi et al. 2016], which models a complex museum with a variety of collections. We note that the model introduced in [Cantino et al. 2007] is an *MDP*. In this case study, we use an *IMDP* model by inserting uncertainties into the *MDP*.

Due to the popularity of the museum, there are many visitors at the same time. Different visitors may have different preferences of arts. We assume the museum divides all collections into different categories so that visitors can choose what they would like to visit and pay tickets according to their preferences. In order to obtain the best experience, a visitor can first assign certain weights to all categories denoting their preferences to the museum, and then design the best strategy for a target. However, the preference of a sort of arts to a visitor may depend on many factors like price, weather, or the length of queue at that moment, etc., hence it is hard to assign fixed values to these preferences. In our model we allow uncertainties of preferences such that their values may lie in an interval.

For simplicity we assume all collections are organised in an $n \times n$ square with $n \geq 10$, with $(0, 0)$ being the south-west corner of the museum and $(n-1, n-1)$ the north-east one. Let $c = \frac{n-1}{2}$; note that (c, c) is at the centre of the museum. We assume all collections at (x, y) are assigned with a weight interval $[3, 4]$ if $\max\{|x-c|, |y-c|\} \leq \frac{n}{10}$, with a weight 2 if $\frac{n}{10} < \max\{|x-c|, |y-c|\} \leq \frac{n}{5}$, and a weight 1 if $\max\{|x-c|, |y-c|\} > \frac{n}{5}$. In other words, we expect collections in the centre to be more popular and subject to more uncertainties than others. Furthermore, we assume that people at each location (x, y) have four nondeterministic choices of moving to (x', y') in the north east, south east, north west, and south west of (x, y) (limited to the boundaries of the museum). The outcome of these choices, however, is not deterministic. That is, deciding to go to (x', y') takes the visitor to either (x, y') or (x', y) depending on the weight intervals of (x, y') and (x', y). Thus, the actual outcome of the move is probabilistic. To obtain an *IMDP*, weights are normalised. For instance, if the visitor chooses to go to the north east and on $(x, y+1)$ there is a weight interval of $[3, 4]$ and on $(x+1, y)$ there is a weight interval of $[2, 2]$, it will go to $(x, y+1)$ with probability interval $[3/(3+2), 4/(4+2)]$ and to $(x+1, y)$ with probability interval $[2/(2+4), 2/(2+3)]$.

Therefore a model with parameter n has n^2 states in total and roughly $4n^2$ transitions, a few of which are associated with uncertain transition probabilities. An instance of the museum model for $n = 14$ is depicted in Fig. 8a. In this instantiation, we assume that the visitor starts in the lower left corner (marked yellow) and wants to move to the upper right corner (marked green) with as few steps as possible. On the other hand, she wants to avoid moving to the black cells, because they correspond to exhibitions which are closed. For closed exhibitions located at $x = 2$, the visitor receive a penalty of 2, for those at $x = 5$ it receives a penalty of 4, for $x = 8$ one of 16 and for $x = 11$ one of 64. Therefore, there is a trade-off between leaving the museum as fast as possible and minimising the penalty received. With r_s being the reward structure for the number of steps and r_p denoting the penalty accumulated, $([\mathsf{r}_s]^{\leq\infty}_{\leq 40}, [\mathsf{r}_p]^{\leq\infty}_{\leq 70})$ requires that we leave the museum within 40 steps but with a penalty of no more than 70. The red arrows indicate a strategy which has been used when computing the Pareto curve by our tool. Here, the visitor mostly ignores closed exhibitions at $x = 2$ but avoids them later. We provide the Pareto curve for this situation in Fig. 8b. With an increasing step bound considered acceptable, the optimal accumulated penalty decreases. This is expected, because with an increasing step bound, the visitor has more time to walk around more of the closed exhibitions, thus facing a lower penalty.

In Fig. 9, we provide strategies for different points on the Pareto curve in Fig. 8b. The lowest expected number of steps in which the museum can be left is 30.9665389. To achieve this number, there is a single optimal strategy sketched in Fig. 9a. As we see, the tourist indeed leaves the museum as soon as possible, by ignoring any closed exhibitions and thus by receiving an expected penalty as high as 152.0609886.

In Fig. 9b and Fig. 9c, we give the tourist somewhat more time, namely 31 steps, so that the penalty of 151.7077821 is a bit lower. Here, with a high probability (0.9894174) the same strategy as for the previous case is chosen. With a probability of 0.0105826 however, the less reckless strategy of Fig. 9c is used, which takes some efforts to avoid the last row of closed exhibitions at $x = 11$.

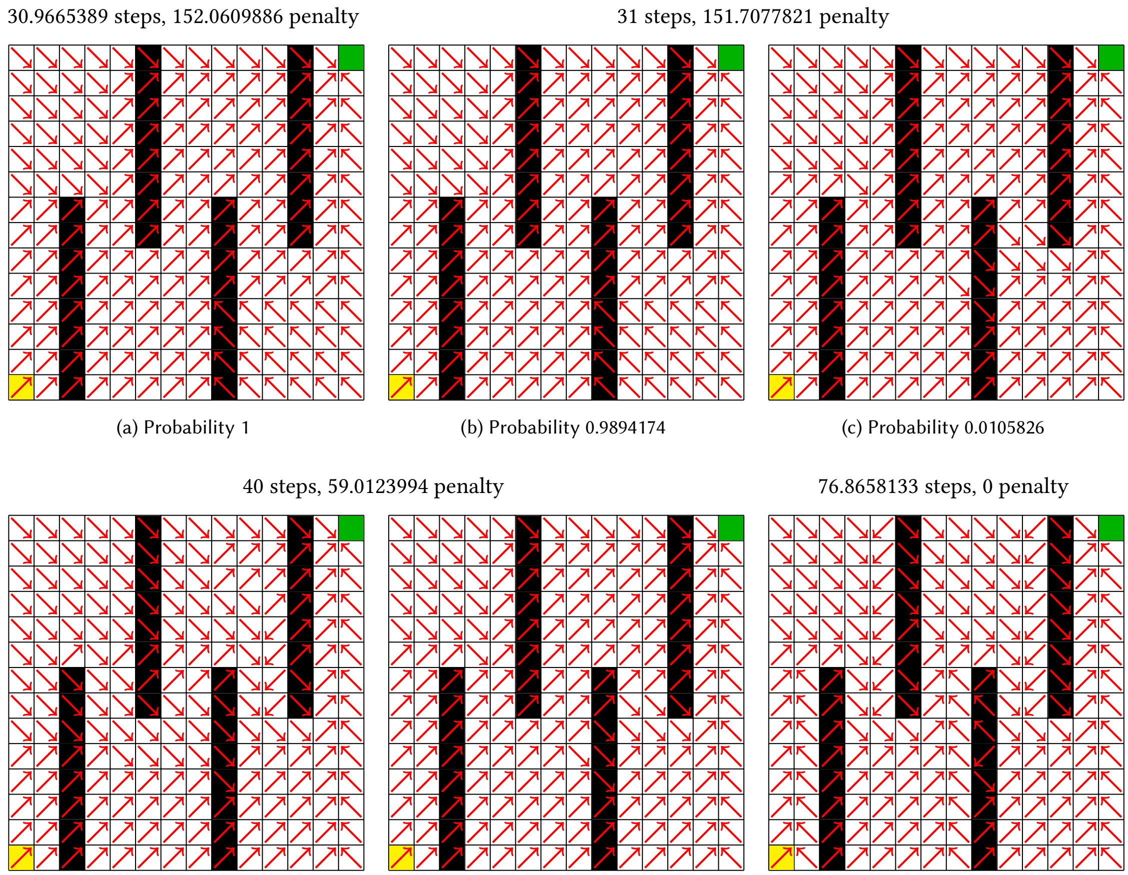
If we further increase the time bound to 40, as in Fig. 9d and Fig. 9e, the strategies used become even less risky but more time consuming to execute.

For a step bound of 76.8658133 and larger, it is possible to avoid receiving any penalty by using the strategy of Fig. 9f, which circumvents all closed exhibitions.

6 CONCLUDING REMARKS

In this paper, we have analysed interval Markov decision processes under controller synthesis semantics in a dynamic setting. In particular, we discussed the problem of multi-objective robust control of *IMDP*s where our goal is to generate an approximation of the Pareto curve for synthesis, quantitative, and Pareto queries. The approximated Pareto curves for various queries include all non-dominated solutions each of which corresponds to a robust strategy that satisfies a given multi-objective predicate under all resolutions of the uncertainty in the transition probabilities. The core part of our approach to approximate Pareto curves of the multi-objective queries was to optimise the weighted sum of objectives which was in turn achieved through a value iteration algorithm. Our designed value iteration algorithm could handle optimising mixture of time bounded and unbounded properties simultaneously which is not the case in standard value iteration algorithms. Additionally, our value iteration algorithm ensures the scalability of our solution methodology compared to linear programming based approaches to optimise the weighted sum of objectives. As we discussed, our proposed approach for optimal control of *IMDP*s with multiple objectives can also be applied to approximate Pareto curves for *MDP*s with convex uncertainty sets as well as ω-regular properties such as PLTL. We finally presented results obtained with a prototype tool on several real-world case studies to show the effectiveness of the developed algorithms.

For future work, we aim to explore the upper bound of the time complexity of the multi-objective robust strategy synthesis problem for *IMDP*s which is left open in this paper.



(a) Probability 1 (b) Probability 0.9894174 (c) Probability 0.0105826

(d) Probability 0.7230247 (e) Probability 0.2769753 (f) Probability 1

Fig. 9. Strategies for different points on the Pareto curve in Fig. 8a.

REFERENCES

Christel Baier and Joost-Pieter Katoen. 2008. *Principles of Model Checking*. The MIT Press.

Nicolas Basset, Marta Kwiatkowska, and Clemens Wiltsche. 2014. Compositional controller synthesis for stochastic games. In *CONCUR*. Springer, 173–187.

Richard Bellman. 1957. A Markovian Decision Process. *Indiana University Mathematics Journal* 6 (1957), 679–684. Issue 4.

Michael Benedikt, Rastislav Lenhardt, and James Worrell. 2013. LTL Model Checking of Interval Markov Chains. In *TACAS*. 32–46.

Dimitris Bertsimas and John N. Tsitsiklis. 1997. *Introduction to Linear Optimization*. Athena Scientific.

Andrea Bianco and Luca de Alfaro. 1995. Model Checking of Probabalistic and Nondeterministic Systems. In *FSTTCS (LNCS)*, Vol. 1026. 499–513.

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Andrew S. Cantino, David L. Roberts, and Charles L. Isbell. 2007. Autonomous nondeterministic tour guides: improving quality of experience with TTD-MDPs. In *AAMAS*. 22.

Krishnendu Chatterjee, Rupak Majumdar, and Thomas A. Henzinger. 2006. Markov Decision Processes with Multiple Objectives. In *STACS (LNCS)*, Vol. 3884. 325–336.

Krishnendu Chatterjee, Koushik Sen, and Thomas A. Henzinger. 2008. Model-Checking ω-Regular Properties of Interval Markov Chains. In *FoSSaCS*. 302–317.

Taolue Chen, Vojtěch Forejt, Marta Kwiatkowska, Aistis Simaitis, and Clemens Wiltsche. 2013a. On stochastic games with multiple objectives. In *MFCS*. Springer, 266–277.

Taolue Chen, Tingting Han, and Marta Kwiatkowska. 2013b. On the complexity of model checking interval-valued discrete time Markov chains. *Inf. Process. Lett.* 113, 7 (2013), 210–216.

Murat Cubuktepe, Nils Jansen, Sebastian Junges, Joost-Pieter Katoen, Ivan Papusha, Hasan A. Poonawala, and Ufuk Topcu. 2017. Sequential Convex Programming for the Efficient Verification of Parametric MDPs. In *TACAS*. 133–150.

Conrado Daws. 2004. Symbolic and Parametric Model Checking of Discrete-Time Markov Chains. In *ICTAC*. 280–294.

Matthias Ehrgott. 2006. *Multicriteria optimization*. Springer Science & Business Media.

Marie-Aude Esteve, Joost-Pieter Katoen, Viet Yen Nguyen, Bart Postma, and Yuri Yushtein. 2012. Formal Correctness, Safety, Dependability and Performance Analysis of a Satellite. In *ICSE*. 1022–1031.

Kousha Etessami, Marta Kwiatkowska, Moshe Y Vardi, and Mihalis Yannakakis. 2007. Multi-objective model checking of Markov decision processes. In *TACAS*. Springer, 50–65.

Harald Fecher, Martin Leucker, and Verena Wolf. 2006. Don't Know in Probabilistic Systems. In *SPIN (LNCS)*, Vol. 3925. Springer, 71–88.

Vojtěch Forejt, Marta Kwiatkowska, Gethin Norman, David Parker, and Hongyang Qu. 2011. Quantitative multi-objective verification for probabilistic systems. In *TACAS*. Springer, 112–127.

Vojtěch Forejt, Marta Kwiatkowska, and David Parker. 2012. Pareto curves for probabilistic model checking. In *ATVA*. Springer, 317–332.

Robert Givan, Sonia M. Leach, and Thomas L. Dean. 2000. Bounded-parameter Markov Decision Processes. *Artif. Intell.* 122, 1-2 (2000), 71–109.

Ernst Moritz Hahn, Tingting Han, and Lijun Zhang. 2011. Synthesis for PCTL in Parametric Markov Decision Processes. In *NFM (LNCS)*, Vol. 6617. 146–161.

Ernst Moritz Hahn, Vahid Hashemi, Holger Hermanns, Morteza Lahijanian, and Andrea Turrini. 2017. Multi-objective Robust Strategy Synthesis for Interval Markov Decision Processes. In *QEST (LNCS)*, Vol. 10503. 207–223.

Vahid Hashemi, Holger Hermanns, and Lei Song. 2016. Reward-Bounded Reachability Probability for Uncertain Weighted MDPs. In *VMCAI*. Springer, 351–371.

Bengt Jonsson and Kim Guldstrand Larsen. 1991. Specification and Refinement of Probabilistic Processes. In *LICS*. IEEE Computer Society, 266–277.

Igor Kozine and Lev V. Utkin. 2002. Interval-Valued Finite Markov Chains. *Reliable Computing* 8, 2 (2002), 97–113.

Marta Kwiatkowska, Gethin Norman, David Parker, and Hongyang Qu. 2013. Compositional probabilistic verification through multi-objective model checking. *Information and Computation* 232 (2013), 38–65.

Morteza Lahijanian, Sean B. Andersson, and Calin Belta. 2015. Formal Verification and Synthesis for Discrete-Time Stochastic Systems. *IEEE Trans. Automat. Control* 60, 8 (2015), 2031–2045.

Morteza Lahijanian and Marta Kwiatkowska. 2016. Specification Revision for Markov Decision Processes with Optimal Trade-off. In *Conf. on Decision and Control*. IEEE, 7411–7418.

Ryan Luna, Morteza Lahijanian, Mark Moll, and Lydia E. Kavraki. 2014a. Asymptotically Optimal Stochastic Motion Planning with Temporal Goals. In *WAFR*. 335–352.

Ryan Luna, Morteza Lahijanian, Mark Moll, and Lydia E. Kavraki. 2014b. Fast Stochastic Motion Planning with Optimality Guarantees using Local Policy Reconfiguration. In *ICRA*. 3013–3019.

Ryan Luna, Morteza Lahijanian, Mark Moll, and Lydia E. Kavraki. 2014c. Optimal and Efficient Stochastic Motion Planning in Partially-Known Environments. In *AAAI*. 2549–2555.

Abdel-Illah Mouaddib. 2004. Multi-objective Decision-theoretic Plan Problem. In *ICRA*, Vol. 3. 2814–2819.

Arnab Nilim and Laurent El Ghaoui. 2005. Robust Control of Markov Decision Processes with Uncertain Transition Matrices. *Operations Research* 53, 5 (2005), 780–798.

Wlodzimierz Ogryczak, Patrice Perny, and Paul Weng. 2013. A Compromise Programming Approach to multiobjective Markov Decision Processes. *IJITDM* 12, 5 (2013), 1021–1054.

Patrice Perny, Paul Weng, Judy Goldsmith, and Josiah P. Hanna. 2013. Approximation of Lorenz-Optimal Solutions in Multiobjective Markov Decision Processes. In *AAAI*. 92–94.

Laure Petrucci and Jaco van de Pol. 2018. Parameter Synthesis Algorithms for Parametric Interval Markov Chains. In *FORTE 2018*. 121–140.

Alberto Puggelli. 2014. *Formal Techniques for the Verification and Optimal Control of Probabilistic Systems in the Presence of Modeling Uncertainties*. Ph.D. Dissertation. EECS Department, University of California, Berkeley. http://www2.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-155.html

Alberto Puggelli, Wenchao Li, Alberto L. Sangiovanni-Vincentelli, and Sanjit A. Seshia. 2013. Polynomial-Time Verification of PCTL Properties of MDPs with Convex Uncertainties. In *CAV*. 527–542.

Tim Quatmann, Christian Dehnert, Nils Jansen, Sebastian Junges, and Joost-Pieter Katoen. 2016. Parameter Synthesis for Markov Models: Faster Than Ever. In *ATVA 2016*. 50–67.

Mickael Randour, Jean-François Raskin, and Ocan Sankur. 2015. Percentile queries in multi-dimensional Markov decision processes. In *CAV*. Springer, 123–139.

Dimitri Scheftelowitsch, Peter Buchholz, Vahid Hashemi, and Holger Hermanns. 2017. Multi-Objective Approaches to Markov Decision Processes with Uncertain Transition Parameters. In *VALUETOOLS 2017*. 44–51.

Eric M. Wolff, Ufuk Topcu, and Richard M. Murray. 2012. Robust control of uncertain Markov Decision Processes with temporal logic specifications. In *CDC*. 3372–3379.

Di Wu and Xenofon D. Koutsoukos. 2008. Reachability analysis of uncertain systems using bounded parameter Markov decision processes. *Artificial Intelligence* 172, 8-9 (2008), 945–954.

A PROOFS OF THE RESULTS ENUNCIATED IN THE PAPER

This appendix contains the proofs of the results enunciated in the main part of the paper.

In order to prove Theorem 17, we need to define the multiple reachability problem for *MDP*s. Formally,

Definition 29. Given an *MDP* M and a reachability predicate described as a vector $\varphi = (\varphi_1, \ldots, \varphi_n)$ where $\varphi_j = [T_j]^{\leq k_j}_{\sim p_j}$ for $j \in \{1, \ldots, n\}$, the multiple reachability problem asks to check if there exists a strategy σ of M such that $\mathsf{M}, \sigma \models \varphi$. The almost-sure multiple reachability problem restricts to $\sim\, = \,\geq$ and $p_j = 1$ for all $j \in \{1, \ldots, n\}$.

The proof makes also use of the following lemma:

Lemma 30 (Complexity of the multi-objective reachability problem for *MDP*s [Randour et al. 2015]). *Given an MDP M, the almost-sure multiple reachability problem is **PSPACE**-complete and strategies need exponential memory in the query size.*

Proof of Theorem 17. We reduce the problem in Lemma 30 to the one under our analysis. In fact, any instance of the multiple reachability problem for *MDP* M can be seen as an instance of the multi-objective robust strategy synthesis problem for an *IMDP* $\mathcal{M}$ generated from M by replacing all probability values with point intervals. Since the multiple reachability problem for *MDP*s is **PSPACE**-complete and the reduction is performed in polynomial time therefore, solving the robust strategy synthesis problem for *IMDP*s is at least **PSPACE**-hard. ◻

Proof of Theorem 22. The proof follows closely the one in [Forejt et al. 2012]. In every iteration of the loop in Algorithm 1, a point $\mathbf{g}$ on a unique face of the Pareto curve is identified. The number of faces of the Pareto curve $\mathcal{P}_{\mathcal{M},\varphi}$ is, in the worst case, exponential in $|\mathcal{M}|$, $\mathbf{k}$, and n [Etessami et al. 2007]. Therefore, termination of Algorithm 1 is guaranteed and the correctness is ensured as a result of the correctness of Algorithm 1 in [Forejt et al. 2012]. The soundness and completeness of the Algorithm 1 is followed by the fact that in every iteration of the algorithm through lines 8-9, the individual model checking problems can be solved in polynomial time in $|\mathcal{M}|$ by formulating the weighted sum of n objectives as a linear programming problem. To see this, without loss of generality, assume that $k_i = \infty$ for all $i \in \{1, \ldots, n\}$. Therefore, following the approach in [Puggelli 2014], the problem of maximising the $ExpTot^{\sigma,\mathbf{k}}_{\mathcal{M}}[\mathbf{w} \cdot \mathsf{r}]$ across the range of strategies $\sigma \in \Sigma$ can be formulated as the following optimisation problem:

$$\begin{aligned}
&\min_{x} \quad \mathbf{x}^T \mathbf{1} \\
&\text{subject to:} \\
&x_s \geq \textstyle\sum_{i=1}^{n} w_i \cdot \mathsf{r}_i(s,a) + \min_{\mathfrak{h}^a_s \in \mathcal{H}^a_s} \mathbf{x}^T \mathfrak{h}^a_s \qquad \forall s \in S, \forall a \in \mathcal{A}(s)
\end{aligned}$$

We now modify the above optimisation problem to simplify derivation of the LP problem. To this aim, we transform the optimisation operator “min” to “max”. Therefore, we get the following optimisation problem:

$$\begin{aligned}
&\max_{x} \quad -\mathbf{x}^T \mathbf{1} \\
&\text{subject to:} \\
&x_s \geq \textstyle\sum_{i=1}^{n} w_i \cdot \mathsf{r}_i(s,a) + \min_{\mathfrak{h}^a_s \in \mathcal{H}^a_s} \mathbf{x}^T \mathfrak{h}^a_s \qquad \forall s \in S, \forall a \in \mathcal{A}(s)
\end{aligned}$$

As it is clear from the set of constraints in the latter optimization problem, the inner optimisation problem is not linear. In order to overcome this difficulty and induce the LP formulation, we follow the techniques in [Puggelli 2014] and use dual of the inner optimisation problem. To this aim, consider the inner optimisation problem with fixed $\mathbf{x}$:

$$P(\mathbf{x}) := \min_{\mathfrak{h}_s^a \in \mathcal{H}_s^a} \mathbf{x}^T \mathfrak{h}_s^a$$

Based on the general description of the interval uncertainty set $\mathcal{H}_s^a = \{\, \mathfrak{h}_s^a \mid \vec{0} \leq \underline{\mathfrak{h}_s^a} \leq \mathfrak{h}_s^a \leq \overline{\mathfrak{h}_s^a} \leq \vec{1}, \mathbf{1}^T \mathfrak{h}_s^a = 1 \,\}$, we can rewrite the latter inner optimisation problem as:

$$\begin{aligned}
&P(\mathbf{x}) := \min \mathbf{x}^T \mathfrak{h}_s^a \\
&\text{subject to:} \\
&\mathbf{1}^T \mathfrak{h}_s^a = 1 \\
&\underline{\mathfrak{h}_s^a} \leq \mathfrak{h}_s^a \leq \overline{\mathfrak{h}_s^a}
\end{aligned}$$

The dual of the above problem is formulated as follows:

$$\begin{aligned}
&D(\mathbf{x}) := \max_{\gamma_{j,1}^{s,a}, \gamma_{j,2}^{s,a}, \gamma_{j,3}^{s,a}} \gamma_{j,1}^{s,a} + \underline{\mathfrak{h}_s^a}^T \gamma_{j,2}^{s,a} - \overline{\mathfrak{h}_s^a}^T \gamma_{j,3}^{s,a} \\
&\text{subject to:} \\
&\mathbf{x} - \gamma_{j,2}^{s,a} + \gamma_{j,3}^{s,a} - \gamma_{j,1}^{s,a} \mathbf{1} = \mathbf{0} \\
&\gamma_{j,2}^{s,a} \geq 0, \gamma_{j,3}^{s,a} \geq 0
\end{aligned}$$

Since the latter inner optimisation problem with fixed $\mathbf{x}$ is an LP, therefore due to the strong duality theorem [Bertsimas and Tsitsiklis 1997], we have $P^*(\mathbf{x}) = D^*(\mathbf{x})$ where $P^*(\mathbf{x})$ and $D^*(\mathbf{x})$ are the primal and dual optimal values, respectively. Therefore, we can replace the original inner optimisation problem with its dual LP to derive the ultimate LP formulation. Note that the inner optimisation operator is removed as the outer optimisation operator will find the least underestimate to maximise its objective function. Hence, maximising the expected total reward for *IMDP* $\mathcal{M}$ with respect to the reward structure $\mathbf{w} \cdot \mathsf{r}$ is formulated as the following LP which can in turn be solved in polynomial time.

$$\begin{aligned}
&\max_{x, \gamma} \quad -\mathbf{x}^T \mathbf{1} \\
&\text{subject to:} \\
&x_s \geq \textstyle\sum_{i=1}^n w_i \cdot \mathsf{r}_i(s,a) + \gamma_{j,1}^{s,a} + \underline{\mathfrak{h}_s^a}^T \gamma_{j,2}^{s,a} - \overline{\mathfrak{h}_s^a}^T \gamma_{j,3}^{s,a} && \forall s \in S, \forall a \in \mathcal{A}(s) \\
&\mathbf{x} - \gamma_{j,2}^{s,a} + \gamma_{j,3}^{s,a} - \gamma_{j,1}^{s,a} \mathbf{1} = \mathbf{0} && \forall s \in S, \forall a \in \mathcal{A}(s) \\
&\gamma_{j,2}^{s,a}, \gamma_{j,3}^{s,a} \geq 0 && \forall s \in S, \forall a \in \mathcal{A}(s)
\end{aligned}$$

□

Proof of Proposition 18. Given a state $(s, v) \in S'$, let $v_e = \{\, i \in \{1, \dots, n\} \mid s \in T_i \,\} \setminus v$. By definition of the transition probability function, it follows that the only successors (s', v') that can be reached from (s, v) must have $v' = v \cup v_e$; moreover, the action performed for such a transition must be of the form (a, v_e). This means that the sets v_e and v' are uniquely determined by the current state (s, v); let $\nu \colon S' \to 2^{\{1,\dots,n\}}$ be the function such that $\nu(s, v) = \{\, i \in \{1, \dots, n\} \mid s \in T_i \,\} \setminus v$ for each $(s, v) \in S'$, $\nu_{\mathcal{A}} \colon S' \times \mathcal{A} \to \mathcal{A}'$ be the function such that $\nu_{\mathcal{A}}((s, v), a) = (a, \nu(s, v))$ for each $(s, v) \in S'$ and $a \in \mathcal{A}$, and $\nu_S \colon S' \times S \to S'$ be the function such that $\nu_S((s, v), s') = (s', v \cup \nu(s, v))$ for each $(s, v) \in S'$ and $s' \in S$.

It is immediate to see that every path ξ' of $\mathcal{M}'$, $\xi' = (s_0, v_0)(a_0, v'_0)(s_1, v_1)(a_1, v'_1)(s_2, v_2)\dots$, is actually of the form $\xi' = (s_0, v_0)\nu_{\mathcal{A}}((s_0, v_0), a_0)(s_1, v_1)\nu_{\mathcal{A}}((s_1, v_1), a_1)(s_2, v_2)\dots$ where $(s_{j+1}, v_{j+1}) = \nu_S((s_j, v_j), s_{j+1})$ for each $j \in \mathbb{N}$, i.e., $v_{j+1} = v_j \cup \nu(s_j, v_j)$. This means that we can define a bijection $\sharp \colon \mathit{Paths} \to \mathit{Paths}'$ as follows: given a path

$\xi = s_0 a_0 s_1 a_1 s_2 \dots$ of $\mathcal{M}$, $\sharp(\xi)$ is defined as $\sharp(\xi) = (s_0, v_0)(a_0, v'_0)(s_1, v_1)(a_1, v'_1)(s_2, v_2)\dots$ where $v_0 = \emptyset$ and for each $j \in \mathbb{N}$, $(a_j, v'_j) = \nu_{\mathcal{A}}((s_j, v_j), a_j)$ and $(s_{j+1}, v_{j+1}) = \nu_S((s_j, v_j), s_j)$.

The inverse $\flat\colon \mathit{Paths}' \to \mathit{Paths}$ of $\sharp$ is just the projection on $\mathcal{M}$: given a path $\xi' = (s_0, v_0)(a_0, v'_0)(s_1, v_1)(a_1, v'_1)(s_2, v_2)\dots$ of $\mathcal{M}'$, $\flat(\xi')$ is defined as $\flat(\xi') = s_0 a_0 s_1 a_1 s_2 \dots$.

Moreover, since the sequence of sets $v_0 v_1 v_2 \dots$ is monotonic non-decreasing with respect to the subset inclusion partial order, we have that, for a given $i \in \{1, \dots, n\}$, if $i \in v_N$ for some $N \in \mathbb{N}$, then there exists exacly one $l \in \mathbb{N}$ such that $i \notin v_j$ for each $0 \leq j < l$ and $i \in v_j$ for each $j \geq l$, i.e., s_l is the first time a state $s \in T_i$ occurs along $\flat(\xi')$. Therefore, it follows that $i \in \nu(s_l, v_l)$ while $i \notin \nu(s_j, v_j)$ for each $j \in \mathbb{N} \setminus \{l\}$. This implies that $\mathsf{r}_{T_i}(\xi'[l], \xi'(l)) = 1$ if $\sim_i = \geq$ or $\mathsf{r}_{T_i}(\xi'[l], \xi'(l)) = -1$ if $\sim_i = \leq$ while $\mathsf{r}_{T_i}(\xi'[j], \xi'(j)) = 0$ for each $j \in \mathbb{N} \setminus \{l\}$, thus

$$\mathsf{r}_{T_i}[k](\xi') = \begin{cases} 1 & \text{if } l < k \text{ and } \sim_i = \geq, \\ -1 & \text{if } l < k \text{ and } \sim_i = \leq, \\ 0 & \text{otherwise.} \end{cases}$$

Note that, if $i \notin v_j$ for each $j \in \mathbb{N}$, then this means that $i \notin \nu(s_j, v_j)$ for each $j \in \mathbb{N}$, thus $\mathsf{r}_{T_i}(\xi'[j], \xi'(j)) = 0$ for each $j \in \mathbb{N}$ and $\mathsf{r}_{T_i}[k](\xi') = 0$.

Similarly, for each $h \in \{n+1, \dots, m\}$, we get that $\bar{\mathsf{r}}_h[k](\xi') = \mathsf{r}_h[k](\xi)$ if $\sim_h = \geq$ and $\bar{\mathsf{r}}_h[k](\xi') = -\mathsf{r}_h[k](\xi)$ if $\sim_h = \leq$.

We are now ready to prove the statement of the proposition, by considering the two implications separately.

Suppose that φ is satisfiable in $\mathcal{M}$: by definition, it follows that there exists a strategy σ of $\mathcal{M}$ such that $\mathcal{M}{\downharpoonright_\sigma} \models_\Pi \varphi$, that is, $\mathcal{M}{\downharpoonright_\sigma} \models_\Pi [T_i]^{\leq k_i}_{\sim_i p_i}$ for each $i \in \{1, \dots, n\}$ and $\mathcal{M}{\downharpoonright_\sigma} \models_\Pi [\mathsf{r}_h]^{\leq k_h}_{\sim_h r_h}$ for each $h \in \{n+1, \dots, m\}$. Let σ' be the strategy of $\mathcal{M}'$ such that, for each finite path $\xi' \in \mathit{FPaths}'$ and action $a \in \mathcal{A}$, $\sigma(\xi')(\nu_{\mathcal{A}}(\mathit{last}(\xi'), a)) = \sigma(\flat(\xi'))(a)$, 0 otherwise. Intuitively, σ' chooses the next action (a, v) exactly as σ chooses a since v is uniquely determined by ξ'. We claim that σ' is such that $\mathcal{M}'{\downharpoonright_{\sigma'}} \models_\Pi \varphi'$.

Let $i \in \{1, \dots, n\}$ and consider $\varphi'_i = [\mathsf{r}_{T_i}]^{\leq k_i+1}_{\geq p'_i}$: there are two cases depending on the original bound $\sim_i$.

If $\sim_i = \geq$, then $[\mathsf{r}_{T_i}]^{\leq k_i+1}_{\geq p'_i} = [\mathsf{r}_{T_i}]^{\leq k_i+1}_{\geq p_i}$; $\mathcal{M}'{\downharpoonright_{\sigma'}} \models_{\Pi'} [\mathsf{r}_{T_i}]^{\leq k_i+1}_{\geq p_i}$ if and only if $\min_{\pi' \in \Pi'} \int_{\xi'} \mathsf{r}_{T_i}[k_i+1](\xi') \,\mathrm{d}\Pr^{\sigma',\pi'}_{\mathcal{M}'} \geq p_i$. Since for each path $\xi' \in \mathit{Paths}'$, $\mathsf{r}_{T_i}[k_i+1](\xi') = 1$ if there exists $l < k_i + 1$ such that $\flat(\xi')[l] \in T_i$, $\mathsf{r}_{T_i}[k_i+1](\xi') = 0$ otherwise, by the way I' and σ' are defined it follows that $\min_{\pi' \in \Pi'} \int_{\xi'} \mathsf{r}_{T_i}[k_i+1](\xi') \,\mathrm{d}\Pr^{\sigma',\pi'}_{\mathcal{M}'} = \min_{\pi \in \Pi} \Pr^{\sigma,\pi}_{\mathcal{M}}\{\, \xi \in \mathit{IPaths} \mid \exists l \leq k : \xi[l] \in T_i \,\}$. Since by hypothesis φ is satisfiable in $\mathcal{M}$, then it follows that $\min_{\pi \in \Pi} \Pr^{\sigma,\pi}_{\mathcal{M}}\{\, \xi \in \mathit{IPaths} \mid \exists l \leq k : \xi[l] \in T_i \,\} \geq p_i$, thus $\min_{\pi' \in \Pi'} \int_{\xi'} \mathsf{r}_{T_i}[k_i+1](\xi') \,\mathrm{d}\Pr^{\sigma',\pi'}_{\mathcal{M}'} \geq p_i$ holds as well, hence $\mathcal{M}'{\downharpoonright_{\sigma'}} \models_{\Pi'} [\mathsf{r}_{T_i}]^{\leq k_i+1}_{\geq p_i} = [\mathsf{r}_{T_i}]^{\leq k_i+1}_{\geq p'_i}$ is satisfied, as required.

Consider now the second case: if $\sim_i = \leq$, then $[\mathsf{r}_{T_i}]^{\leq k_i+1}_{\geq p'_i} = [\mathsf{r}_{T_i}]^{\leq k_i+1}_{\geq -p_i}$; $\mathcal{M}'{\downharpoonright_{\sigma'}} \models_{\Pi'} [\mathsf{r}_{T_i}]^{\leq k_i+1}_{\geq -p_i}$ if and only if $\min_{\pi' \in \Pi'} \int_{\xi'} \mathsf{r}_{T_i}[k_i+1](\xi') \,\mathrm{d}\Pr^{\sigma',\pi'}_{\mathcal{M}'} \geq -p_i$. Since for each path $\xi' \in \mathit{Paths}'$, $\mathsf{r}_{T_i}[k_i+1](\xi') = -1$ if there exists $l < k_i + 1$ such that $\flat(\xi')[l] \in T_i$, $\mathsf{r}_{T_i}[k_i+1](\xi') = 0$ otherwise, by the way I' and σ' are defined it follows that $\min_{\pi' \in \Pi'} \int_{\xi'} \mathsf{r}_{T_i}[k_i+1](\xi') \,\mathrm{d}\Pr^{\sigma',\pi'}_{\mathcal{M}'} = -\max_{\pi \in \Pi} \Pr^{\sigma,\pi}_{\mathcal{M}}\{\, \xi \in \mathit{IPaths} \mid \exists l \leq k : \xi[l] \in T_i \,\}$. Since by hypothesis we have that φ is satisfiable in $\mathcal{M}$, then it follows that $\max_{\pi \in \Pi} \Pr^{\sigma,\pi}_{\mathcal{M}}\{\, \xi \in \mathit{IPaths} \mid \exists l \leq k : \xi[l] \in T_i \,\} \leq p_i$, thus $\min_{\pi' \in \Pi'} \int_{\xi'} \mathsf{r}_{T_i}[k_i+1](\xi') \,\mathrm{d}\Pr^{\sigma',\pi'}_{\mathcal{M}'} \geq -p_i$ holds as well, hence $\mathcal{M}'{\downharpoonright_{\sigma'}} \models_{\Pi'} [\mathsf{r}_{T_i}]^{\leq k_i+1}_{\geq -p_i} = [\mathsf{r}_{T_i}]^{\leq k_i+1}_{\geq p'_i}$ is satisfied, as required.

This completes the analysis of the case $\varphi'_i = [\mathsf{r}_{T_i}]^{\leq k_i+1}_{\geq p'_i}$ for each $i \in \{1, \dots, n\}$.

Let $h \in \{n+1, \dots, m\}$ and consider $\varphi'_h = [\bar{\mathsf{r}}_h]^{\leq k_h}_{\geq r'_h}$: there are two cases depending on the original bound $\sim_h$.

If $\sim_h = \geq$, then $[\bar{\mathsf{r}}_h]^{\leq k_h}_{\geq r'_h} = [\bar{\mathsf{r}}_h]^{\leq k_h}_{\geq r_h}$; $\mathcal{M}'\downharpoonright_{\sigma'} \models_{\Pi'} [\bar{\mathsf{r}}_h]^{\leq k_h}_{\geq r_h}$ holds if and only if $\min_{\pi' \in \Pi'} \int_{\xi'} \bar{\mathsf{r}}_h[k_h](\xi') \,\mathrm{d}\mathrm{Pr}^{\sigma',\pi'}_{\mathcal{M}'} \geq r_h$ holds. Since for each path $\xi' \in \mathit{Paths}'$, $\bar{\mathsf{r}}_h[k](\xi') = \mathsf{r}_h[k](\flat(\xi'))$, by the way the components I', $\bar{\mathsf{r}}_h$, and σ' are defined it follows that $\min_{\pi' \in \Pi'} \int_{\xi'} \bar{\mathsf{r}}_h[k_h](\xi') \,\mathrm{d}\mathrm{Pr}^{\sigma',\pi'}_{\mathcal{M}'} = \min_{\pi \in \Pi} \int_{\xi} \mathsf{r}_h[k_h](\xi) \,\mathrm{d}\mathrm{Pr}^{\sigma,\pi}_{\mathcal{M}}$. Since by hypothesis φ is satisfiable in $\mathcal{M}$, then it follows that $\min_{\pi \in \Pi} \int_{\xi} \mathsf{r}_h[k_h](\xi) \,\mathrm{d}\mathrm{Pr}^{\sigma,\pi}_{\mathcal{M}} \geq r_h$, thus $\min_{\pi' \in \Pi'} \int_{\xi'} \bar{\mathsf{r}}_h[k_h](\xi') \,\mathrm{d}\mathrm{Pr}^{\sigma',\pi'}_{\mathcal{M}'} \geq r_h$ holds as well, hence $\mathcal{M}'\downharpoonright_{\sigma'} \models_{\Pi'} [\bar{\mathsf{r}}_h]^{\leq k_h}_{\geq r_h} = [\bar{\mathsf{r}}_h]^{\leq k_h}_{\geq r'_h}$ is satisfied, as required.

Consider now the second case: if $\sim_h = \leq$, then $[\bar{\mathsf{r}}_h]^{\leq k_h}_{\geq r'_h} = [\bar{\mathsf{r}}_h]^{\leq k_h}_{\geq -r_h}$; $\mathcal{M}'\downharpoonright_{\sigma'} \models_{\Pi'} [\bar{\mathsf{r}}_h]^{\leq k_h}_{\geq -r_h}$ if and only if $\min_{\pi' \in \Pi'} \int_{\xi'} \bar{\mathsf{r}}_h[k_h](\xi') \,\mathrm{d}\mathrm{Pr}^{\sigma',\pi'}_{\mathcal{M}'} \geq -r_h$. Since for each path $\xi' \in \mathit{Paths}'$, $\bar{\mathsf{r}}_h[k](\xi') = -\mathsf{r}_h[k](\flat(\xi'))$, by the way I', $\bar{\mathsf{r}}_h$, and σ' are defined it follows that $\min_{\pi' \in \Pi'} \int_{\xi'} \bar{\mathsf{r}}_h[k_h](\xi') \,\mathrm{d}\mathrm{Pr}^{\sigma',\pi'}_{\mathcal{M}'} = -\max_{\pi \in \Pi} \int_{\xi} \mathsf{r}_h[k_h](\xi) \,\mathrm{d}\mathrm{Pr}^{\sigma,\pi}_{\mathcal{M}}$. Since by hypothesis φ is satisfiable in $\mathcal{M}$, then it follows that $\max_{\pi \in \Pi} \int_{\xi} \mathsf{r}_h[k_h](\xi) \,\mathrm{d}\mathrm{Pr}^{\sigma,\pi}_{\mathcal{M}} \leq r_h$, thus $\min_{\pi' \in \Pi'} \int_{\xi'} \bar{\mathsf{r}}_h[k_h](\xi') \,\mathrm{d}\mathrm{Pr}^{\sigma',\pi'}_{\mathcal{M}'} \geq -r_h$ holds as well, hence $\mathcal{M}'\downharpoonright_{\sigma'} \models_{\Pi'} [\bar{\mathsf{r}}_h]^{\leq k_h}_{\geq -r_h} = [\bar{\mathsf{r}}_h]^{\leq k_h}_{\geq r'_h}$ is satisfied, as required.

This completes the analysis of the case $\varphi'_h = [\bar{\mathsf{r}}_h]^{\leq k_h}_{\geq r'_h}$ for each $h \in \{n+1, \dots, m\}$; since $\mathcal{M}'\downharpoonright_{\sigma'} \models_{\Pi'} \varphi'_j$ for each $j \in \{1, \dots, m\}$, it follows that φ is satisfiable in $\mathcal{M}'$, as required to prove that "if φ is satisfiable in $\mathcal{M}$, then φ' is satisfiable in $\mathcal{M}'$".

Suppose now the other implication, namely "if φ' is satisfiable in $\mathcal{M}'$, then φ is satisfiable in $\mathcal{M}$" and assume that φ' is satisfiable in $\mathcal{M}'$: by definition, it follows that there exists a strategy σ' of $\mathcal{M}'$ such that $\mathcal{M}'\downharpoonright_{\sigma'} \models_{\Pi'} \varphi'$, that is, $\mathcal{M}'\downharpoonright_{\sigma'} \models_{\Pi'} [\mathsf{r}_{T_i}]^{\leq k_i+1}_{\geq p'_i}$ for each $i \in \{1, \dots, n\}$ and $\mathcal{M}'\downharpoonright_{\sigma'} \models_{\Pi'} [\bar{\mathsf{r}}_h]^{\leq k_h}_{\geq r'_h}$ for each $h \in \{n+1, \dots, m\}$. Let σ be the strategy of $\mathcal{M}$ such that, for each finite path $\xi \in \mathit{FPaths}$ and action $a \in \mathcal{A}$, $\sigma(\xi)(a) = \sigma'(\sharp(\xi))(a, v)$, 0 otherwise, where $(a, v) = v_{\mathcal{A}}(\mathit{last}(\sharp(\xi)), a)$. Intuitively, σ chooses the next action a exactly as σ' chooses (a, v) since v is uniquely determined by ξ'. We claim that σ is such that $\mathcal{M}\downharpoonright_{\sigma} \models_{\Pi} \varphi$.

Let $i \in \{1, \dots, n\}$ and consider $\varphi_i = [T_i]^{\leq k_i}_{\sim_i p_i}$: there are two cases depending on the bound $\sim_i$.

If $\sim_i = \geq$, then $\mathcal{M}\downharpoonright_{\sigma} \models_{\Pi} [T_i]^{\leq k_i}_{\geq p_i}$ if and only if $\min_{\pi \in \Pi} \mathrm{Pr}^{\sigma,\pi}_{\mathcal{M}}\{\, \xi \in \mathit{IPaths} \mid \exists l \leq k : \xi[l] \in T_i \,\} \geq p_i$. Since for each path $\xi \in \mathit{Paths}$, $\mathsf{r}_{T_i}[k_i+1](\sharp(\xi)) = 1$ if there exists $l < k_i + 1$ such that $\xi[l] \in T_i$, $\mathsf{r}_{T_i}[k_i+1](\sharp(\xi)) = 0$ otherwise, by the way I' and σ are defined it follows that $\min_{\pi \in \Pi} \mathrm{Pr}^{\sigma,\pi}_{\mathcal{M}}\{\, \xi \in \mathit{IPaths} \mid \exists l \leq k : \xi[l] \in T_i \,\} = \min_{\pi' \in \Pi'} \int_{\xi'} \mathsf{r}_{T_i}[k_i + 1](\xi') \,\mathrm{d}\mathrm{Pr}^{\sigma',\pi'}_{\mathcal{M}'}$. Since by hypothesis φ' is satisfiable in $\mathcal{M}'$, then it follows that $\min_{\pi' \in \Pi'} \int_{\xi'} \mathsf{r}_{T_i}[k_i+1](\xi') \,\mathrm{d}\mathrm{Pr}^{\sigma',\pi'}_{\mathcal{M}'} \geq p_i$, thus $\min_{\pi \in \Pi} \mathrm{Pr}^{\sigma,\pi}_{\mathcal{M}}\{\, \xi \in \mathit{IPaths} \mid \exists l \leq k : \xi[l] \in T_i \,\} \geq p_i$ holds as well, hence $\mathcal{M}\downharpoonright_{\sigma} \models_{\Pi} [T_i]^{\leq k_i}_{\geq p_i} = [T_i]^{\leq k_i}_{\sim_i p_i}$ is satisfied, as required.

Consider now the second case: If $\sim_i = \leq$, then $\mathcal{M}\downharpoonright_{\sigma} \models_{\Pi} [T_i]^{\leq k_i}_{\leq p_i}$ if and only if $\max_{\pi \in \Pi} \mathrm{Pr}^{\sigma,\pi}_{\mathcal{M}}\{\, \xi \in \mathit{IPaths} \mid \exists l \leq k : \xi[l] \in T_i \,\} \leq p_i$. Since for each path $\xi \in \mathit{Paths}$, $\mathsf{r}_{T_i}[k_i+1](\sharp(\xi)) = -1$ if there exists $l < k_i + 1$ such that $\xi[l] \in T_i$, $\mathsf{r}_{T_i}[k_i+1](\sharp(\xi)) = 0$ otherwise, by the way I' and σ are defined it follows that $\max_{\pi \in \Pi} \mathrm{Pr}^{\sigma,\pi}_{\mathcal{M}}\{\, \xi \in \mathit{IPaths} \mid \exists l \leq k : \xi[l] \in T_i \,\} = -\min_{\pi' \in \Pi'} \int_{\xi'} \mathsf{r}_{T_i}[k_i+1](\xi') \,\mathrm{d}\mathrm{Pr}^{\sigma',\pi'}_{\mathcal{M}'}$. Since by hypothesis φ' is satisfiable in $\mathcal{M}'$, then it follows that $\min_{\pi' \in \Pi'} \int_{\xi'} \mathsf{r}_{T_i}[k_i+1](\xi') \,\mathrm{d}\mathrm{Pr}^{\sigma',\pi'}_{\mathcal{M}'} \geq -p_i$, thus $\max_{\pi \in \Pi} \mathrm{Pr}^{\sigma,\pi}_{\mathcal{M}}\{\, \xi \in \mathit{IPaths} \mid \exists l \leq k : \xi[l] \in T_i \,\} \leq p_i$ holds as well, hence $\mathcal{M}\downharpoonright_{\sigma} \models_{\Pi} [T_i]^{\leq k_i}_{\leq p_i} = [T_i]^{\leq k_i}_{\sim_i p_i}$ is satisfied, as required.

This completes the analysis of the case $\varphi_i = [T_i]^{\leq k_i}_{\sim_i p_i}$ for each $i \in \{1, \dots, n\}$.

Let $h \in \{n+1, \dots, m\}$ and consider $\varphi_h = [\mathsf{r}_h]^{\leq k_h}_{\sim_h r_h}$: there are two cases depending on the original bound $\sim_h$.

If $\sim_h = \geq$, then $\mathcal{M}\downharpoonright_{\sigma} \models_{\Pi} [\mathsf{r}_h]^{\leq k_h}_{\geq r_h}$ if and only if $\min_{\pi \in \Pi} \int_{\xi} \mathsf{r}_h[k_h](\xi) \,\mathrm{d}\mathrm{Pr}^{\sigma,\pi}_{\mathcal{M}} \geq r_h$. Since for each path $\xi \in \mathit{Paths}$, $\bar{\mathsf{r}}_h[k](\sharp(\xi)) = \mathsf{r}_h[k](\xi)$, by the way I', $\bar{\mathsf{r}}_h$, and σ are defined it follows that $\min_{\pi \in \Pi} \int_{\xi} \mathsf{r}_h[k_h](\xi) \,\mathrm{d}\mathrm{Pr}^{\sigma,\pi}_{\mathcal{M}} =$

$\min_{\pi' \in \Pi'} \int_{\xi'} \bar{r}_h[k_h](\xi') \, \mathrm{dPr}^{\sigma',\pi'}_{\mathcal{M}'}$. Since by hypothesis φ' is satisfiable in $\mathcal{M}'$, then $\min_{\pi' \in \Pi'} \int_{\xi'} \bar{r}_h[k_h](\xi') \, \mathrm{dPr}^{\sigma',\pi'}_{\mathcal{M}'} \geq r_h$, thus $\min_{\pi \in \Pi} \int_{\xi} r_h[k_h](\xi) \, \mathrm{dPr}^{\sigma,\pi}_{\mathcal{M}} \geq r_h$ holds as well, hence $\mathcal{M}\lfloor_\sigma \models_\Pi [r_h]^{\leq k_h}_{\geq r_h} = [r_h]^{\leq k_h}_{\sim_h r_h}$ is satisfied, as required.

Consider now the second case: if $\sim_h = \leq$, then $\mathcal{M}\lfloor_\sigma \models_\Pi [r_h]^{\leq k_h}_{\leq r_h}$ if and only if $\max_{\pi \in \Pi} \int_{\xi} r_h[k_h](\xi) \, \mathrm{dPr}^{\sigma,\pi}_{\mathcal{M}} \leq r_h$. Since for each path $\xi \in \mathit{Paths}$, $-\bar{r}_h[k](\sharp(\xi)) = r_h[k](\xi)$, by the definition of the components I', $\bar{r}_h$, and σ it is the case that $\max_{\pi \in \Pi} \int_{\xi} r_h[k_h](\xi) \, \mathrm{dPr}^{\sigma,\pi}_{\mathcal{M}} = -\min_{\pi' \in \Pi'} \int_{\xi'} \bar{r}_h[k_h](\xi') \, \mathrm{dPr}^{\sigma',\pi'}_{\mathcal{M}'}$. Since by hypothesis φ' is satisfiable in $\mathcal{M}'$, then $\min_{\pi' \in \Pi'} \int_{\xi'} \bar{r}_h[k_h](\xi') \, \mathrm{dPr}^{\sigma',\pi'}_{\mathcal{M}'} \geq -r_h$, thus $\max_{\pi \in \Pi} \int_{\xi} r_h[k_h](\xi) \, \mathrm{dPr}^{\sigma,\pi}_{\mathcal{M}} \leq r_h$ holds as well, hence $\mathcal{M}\lfloor_\sigma \models_\Pi [r_h]^{\leq k_h}_{\leq r_h} = [r_h]^{\leq k_h}_{\sim_h r_h}$ is satisfied, as required.

This completes the analysis of the case $\varphi_h = [r_h]^{\leq k_h}_{\sim_h r_h}$ for each $h \in \{n+1, \dots, m\}$; since $\mathcal{M}\lfloor_\sigma \models_\Pi \varphi_j$ for each $j \in \{1, \dots, m\}$, it follows that φ is satisfiable in $\mathcal{M}$, as required to prove that "if φ' is satisfiable in $\mathcal{M}'$, then φ is satisfiable in $\mathcal{M}$". Having proved both implications, the statement of the proposition "φ is satisfiable in $\mathcal{M}$ if and only if φ' is satisfiable in $\mathcal{M}'$" holds, as required. □

Proof of Proposition 15. We prove this proposition by adapting the proof from [Forejt et al. 2011, Proposition 1].

Direction $\Rightarrow$. Assume that, for a reward structure r, $\sup\{ \mathit{ExpTot}^{\sigma,\infty}_{\mathcal{M}}[r] \mid \mathcal{M}\lfloor_\sigma \models_\Pi ([T_1]^{\leq k_1}_{\sim p_1}, \dots, [T_n]^{\leq k_n}_{\sim p_n}) \} = \infty$. From Lemma 14, it follows that if state-action pair (s, a) occurs infinitely often, s and a are contained in a SEC $E_{\mathcal{M}}$. Therefore, to satisfy the assumed condition, there must exist some strategy σ such that $\mathcal{M}\lfloor_\sigma \models_\Pi ([T_1]^{\leq k_1}_{\sim p_1}, \dots, [T_n]^{\leq k_n}_{\sim p_n})$ and a SEC is reachable, in which σ picks action a at reachable state s with positive probability, and $r(s, a) > 0$.

Direction $\Leftarrow$. Assume that there is a strategy σ such that $\mathcal{M}\lfloor_\sigma \models_\Pi ([T_1]^{\leq k_1}_{\sim p_1}, \dots, [T_n]^{\leq k_n}_{\sim p_n})$, a SEC $E_{\mathcal{M}} = (S', \mathcal{A}')$ is reachable, and $r(\xi[n], \xi(n)) > 0$, where ξ is a finite path of length $n+1$ under σ with $\xi[n] \in S'$ and $\xi(n) \in \mathcal{A}'(\xi[n])$ for some $n \geq 0$. To complete the proof, it is enough to show that there is a sequence of strategies $\{\sigma_k\}_{k \in \mathbb{N}}$ under which (i) the probabilistic predicates $[T_1]^{\leq k_1}_{\sim p_1}, \dots, [T_n]^{\leq k_n}_{\sim p_n}$ are satisfied and (ii) $\lim_{k \to \infty} \mathit{ExpTot}^{\sigma_k,k}_{\mathcal{M}}[r] = \infty$.

(i) Let $\xi[n] = s$ and $\xi(n) = a$. For $k \in \mathbb{N}$ consider σ_k that

- for the paths that do not have the prefix ξ, σ_k emulates σ.
- when the path ξ is performed, σ_k forces the system to stay in $E_{\mathcal{M}}$ containing (s, a). After k occurrences of (s, a), the next time s is visited, the strategy σ_k emulates σ again as if the performed path segment after $\xi[n]$ was never executed.

Under σ_k, the reachability predicates are satisfied for any $k \in \mathbb{N}$. To see this, consider θ_k that maps each path ξ of σ to the paths of σ_k. We now have $\theta(\xi) \cap \theta(\xi') = \emptyset$ for all $\xi \neq \xi'$, and for all sets Ω and two natures π and π_k, where π_k emulates π the same way σ_k emulates σ, we have $\mathrm{Pr}^{\sigma,\pi}_{\mathcal{M}}(\Omega) = \mathrm{Pr}^{\sigma_k,\pi_k}_{\mathcal{M}}(\theta(\Omega))$, independent of the choice of π_k during the execution of the path segment that σ_k forces the stay in $E_{\mathcal{M}}$. The satisfaction of the reachability predicates under each σ_k follows from the fact that, for any path ξ of σ, ξ satisfies a reachability predicate iff each path in $\theta(\Omega)$ satisfies the reachability predicate.

(ii) To show that $\lim_{k \to \infty} \mathit{ExpTot}^{\sigma_k,k}_{\mathcal{M}}[r] = \infty$, recall that the probability of reaching (s, a) under σ_k for the first time is some positive value p_1. From the properties of SEC, the probability of returning to s within l steps, where $l = |S|$, is also some positive value p_2. By construction, (s, a) is picked k times, therefore, $\mathit{ExpTot}^{\sigma_k,k}_{\mathcal{M}}[r] \geq p_1 p_2 \frac{k}{l} r(s, a)$, and hence, $\lim_{k \to \infty} \mathit{ExpTot}^{\sigma_k,k}_{\mathcal{M}}[r] = \infty$. □