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


Published in:
ACM Transactions on Modeling and Computer Simulation

Document Version:
Publisher's PDF, also known as Version of record

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

Publisher rights
Copyright 2019 the authors. This is an open access article published under a Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution and reproduction in any medium, provided the author and source are cited.

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Download date: 25. Jul. 2022
Interval Markov Decision Processes with Multiple Objectives: From Robust Strategies to Pareto Curves

ERNST MORITZ HAHN, The School of Electronics, Electrical Engineering and Computer Science, Queen’s University Belfast, UK and State Key Laboratory of Computer Science, Institute of Software, CAS, China
VAHID HASHEMI, Department of Information Technology, Audi AG, Germany
HOLGER HERMANNS, Saarland University, Saarland Informatics Campus, Germany and Institute of Intelligent Software, Guangzhou, China
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 article, we consider Interval Markov decision processes (IMDPs), which generalise classical MDPs 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 IMDPs and Pareto curve analysis of multi-objective queries on IMDPs. We study how to find a robust (randomised) strategy that satisfies multiple objectives involving rewards, reachability, and more general $\omega$-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

This work is supported by the ERC Advanced Investigators Grant 695614 (POWVER), by the German Research Foundation (DFG) Grant 389792660, as part of CRC 248 (see https://perspicuous-computing.science), 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 (Grant Nos. 61550110506, 61650410658, 6176136011, and 61532019), by the Chinese Academy of Sciences Fellowship for International Young Scientists, by H2020 Marie Skłodowska-Curie Actions Individual Fellowship “PaVeCo” - Parametrised Verification and Control, by the CDZ project CAP (GZ 1023), by Guangdong Science and Technology Department (Grant no. 2018B010107004), and by DFG grant 389792660 as part of https://perspicuous-computing.science TRR-248.

Authors’ addresses: E. M. Hahn, The School of Electronics, Electrical Engineering and Computer Science, Queen’s University Belfast, Belfast, UK, State Key Laboratory of Computer Science, Institute of Software, CAS, Beijing, China; email: e.hahn@qub.ac.uk; V. Hashemi, Department of Information Technology, Audi AG, Ingolstadt, Germany; email: vahid.hashemi@audi.de; H. Hermanns, Saarland University, Saarland Informatics Campus, Saarbrücken, Germany, Institute of Intelligent Software, Guangzhou, Guangzhou, China; email: hermanns@cs.uni-saarland.de; M. Lahijanian, Department of Smead Aerospace Engineering and Sciences, University of Colorado, Boulder, CO; email: morteza.lahijanian@colorado.edu; A. Turrini, Institute of Intelligent Software, Guangzhou, Guangzhou, China, State Key Laboratory of Computer Science, Institute of Software, CAS, Beijing, China; email: turrini@ios.ac.cn.

© 2019 Copyright held by the owner/author(s).
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

ACM Reference format:

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, 2014b, 2014c]; 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 \( p_{\text{safe}} \), the probability of successfully completing a track by safely manoeuvring between obstacles, while minimising \( t_{\text{travel}} \), the total expected travel time.” This example has competing objectives: maximising \( p_{\text{safe}} \), which requires the robot to be conservative, and minimising \( t_{\text{travel}} \), 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 \( p_{\text{safe}} \) must induce an increase in the value of \( t_{\text{travel}} \), 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 article, 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 contrary 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 that optimises the weighted sum of objectives. We consider three different multi-objective queries for IMDPs, 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. 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, we discuss approximating Pareto curves for IMDP models that 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 MDPs. 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 article, 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.

First, from the modelling viewpoint, various probabilistic modelling formalisms with uncertain transitions have been studied in the literature. Interval Markov Chains (IMCs) [Jonsson and Larsen 1991; Kozine and Utkin 2002] or abstract Markov chains [Fecher et al. 2006] extend standard discrete-time Markov Chains (MCs) with interval uncertainties. They do not feature the non-deterministic choices of transitions. Uncertain MDPs [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 MDPs [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. IMDPs extend IMCs by inclusion of nondeterminism and are a subset of uncertain MDPs and parametric MDPs.

Second, 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 IMCs and IMDPs was first investigated in Chen et al. [2013b] and 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 of Hahn et al. [2011] and Nilim and El Ghaoui [2005] considered synthesis for parametric MDPs and MDPs 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 MDPs. 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 IMDPs 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 article. We start with necessary preliminaries in Section 2. In Section 3, we discuss multi-objective robust control of IMDPs 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. In Section 6, we conclude the article.

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

2 PRELIMINARIES

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

For a vector $x \in \mathbb{R}^n$, we denote by $x_i$, its $i$th component, and we call $x$ a weight vector if $x_i \geq 0$ for all $i$ and $\sum_{i=1}^n x_i = 1$. The Euclidean inner product $x \cdot y$ of two vectors $x, y \in \mathbb{R}^n$ 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., $x \leq y$ means that for all indices $i$, we have $x_i \leq y_i$. For a set of vectors $S = \{s_1, \ldots, s_l\} \subseteq \mathbb{R}^n$, we say that $s \in \mathbb{R}^n$ is a convex combination of elements of $S$, if $s = \sum_{i=1}^l w_i \cdot s_i$ for some weight vector $w \in \mathbb{R}^l_\geq$. Furthermore, we denote by $S \downarrow$ the downward closure of the convex hull of $S$ that is defined as $S \downarrow = \{y \in \mathbb{R}^n \mid y \leq z$ for some convex combination $z$ of the elements of $S\}$. For a given convex set $X$, we say that a point $x \in X$ is on the boundary of $X$, denoted by $x \in \partial X$, if for every $\varepsilon > 0$ there is a point $y \notin X$ such that the Euclidean distance between $x$ and $y$ is at most $\varepsilon$. Given a downward closed set $X \subseteq \mathbb{R}^n$, for any $z \in \mathbb{R}^n$ such that $z \in \partial X$ or $z \notin X$, there is a weight vector $w \in \mathbb{R}^n$ such that $w \cdot z \leq w \cdot x$ for all $x \in X$ [Boyd and Vandenberghe 2004]. We say that $w$ separates $z$ from $X \downarrow$. Given a set $Y \subseteq \mathbb{R}^k$, we call a vector $y \in Y$ Pareto optimal in $Y$ if there does not exist a vector $z \in Y$ such that $y \leq z$ and $y \neq z$. 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 $\mathcal{Y} = \{y \in Y \mid y$ is Pareto optimal $\}$. 

2.1 Interval Markov Decision Processes

We now define Interval Markov Decision Processes (IMDP) as an extension of MDPs, which allow for the inclusion of transition probability uncertainties as intervals. IMDPs belong to the family of uncertain MDPs and allow to describe a set of MDPs 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, $I : S \times \mathcal{A} \times S \rightarrow [0, \infty]$ is a total interval transition probability function where $I = \{[a, b] \mid 0 < a \leq b \leq 1\}$, $AP$ if a finite set of atomic propositions, and $L : S \rightarrow 2^{AP}$ 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 IMDPs. Given $s \in S$ and $a \in \mathcal{A}$, we call $b^s_a \in \text{Disc}(S)$ a feasible distribution reachable from $s$ by $a$, denoted by $s \xrightarrow{a} b^s_a$, if, for each state $s' \in S$, we have $b^s_a(s') \in I(s, a, s')$. 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 $\mathcal{H}^s_a$, i.e., $\mathcal{H}^s_a = \{ b^s_a \in \text{Disc}(S) \mid s \xrightarrow{a} b^s_a \}$ and we denote the set of available actions at state $s \in S$ by $\mathcal{A}(s)$, i.e., $\mathcal{A}(s) = \{ a \in \mathcal{A} \mid \mathcal{H}^s_a \neq \emptyset \}$. 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., $|\mathcal{M}| = |\{(s, a, s', i) \in S \times \mathcal{A} \times S \times I \mid I(s, a, s') = i\}| \in O(|S|^2 \cdot |\mathcal{A}|)$.

A path $\xi$ in $\mathcal{M}$ is a finite or infinite sequence of alternating states and actions $\xi = s_0a_0s_1 \ldots$, ending with a state if finite, such that for each $i \geq 0$, $I(s_i, a_i, s_{i+1}) \in I$. The $i$th state (action) along the path $\xi$ 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 $\xi[i \ldots]$ the suffix of $\xi$ starting from $\xi[i]$. For instance, for the finite path $\xi = s_0a_0s_1 \ldots s_n$, 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 $\text{FPaths}$ and $\text{IPaths}$, respectively.

An $\omega$-word $w$ is an infinite sequence of sets of atomic propositions, i.e., $w \in (2^{AP})^\omega$. Given an infinite path $\xi$, the word $w(\xi)$ generated by $\xi$ is the sequence $w(\xi) = w_0w_1 \ldots$ 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 $\sigma : \text{FPaths} \rightarrow \text{Disc}(\mathcal{A})$ such that for each $\xi \in \text{FPaths}$, $\sigma(\xi) \in \text{Disc}(\mathcal{A}(\text{last}(\xi)))$. A nature is a function $\pi : \text{FPaths} \times \mathcal{A} \rightarrow \text{Disc}(\mathcal{A})$ such that for each $\xi \in \text{FPaths}$ and $a \in \mathcal{A}(s)$, $\pi(\xi, a) \in \mathcal{H}^s_a$ where $s =$ last($\xi$). The sets of all strategies and all natures are denoted by $\Sigma$ and $\Pi$, respectively.

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

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

\[
\Pr^\sigma_\pi(\text{Cyl}_{\xi}) = \begin{cases} 
1 & \text{if } \xi = \bar{s}, \\
0 & \text{if } \xi = t \neq \bar{s}, \\
\Pr^\sigma_\pi(\text{Cyl}_{\xi'}) \cdot \sigma(\xi'(a)) \cdot \pi(\xi', a)(s) & \text{if } \xi = \xi' as. 
\end{cases}
\]
Standard measure theoretical arguments ensure that $\Pr_{\mathcal{M}}^{\sigma,\pi}$ extends uniquely to the $\sigma$-field generated by cylinder sets.

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 $r : S \times \mathcal{A} \to \mathbb{R}$ that assigns to each state-action pair $(s, a)$, where $s \in S$ and $a \in \mathcal{A}(s)$, a reward $r(s, a) \in \mathbb{R}$. Given a path $\xi$ and $k \in \mathbb{N} \cup \{\infty\}$, the total accumulated reward in $k$ steps for $\xi$ over $r$ is $r[k](\xi) = \sum_{i=0}^{k-1} r(\xi[i], \xi(i))$.

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 an IMDP with a reward structure, consider the IMDP depicted in Figure 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 $I(s, a, t) = \left[\frac{1}{3}, \frac{2}{3}\right]$, $I(s, a, u) = \left[\frac{1}{5}, 1\right]$, $I(s, b, t) = \left[\frac{2}{5}, \frac{3}{5}\right]$, $I(s, b, u) = \left[\frac{1}{3}, \frac{2}{3}\right]$, and $I(t, a, t) = I(u, b, u) = \{1, 1\}$. The underlined numbers indicate the reward structure $r$ with $r(s, a) = 3$, $r(s, b) = 1$, and $r(t, a) = r(u, b) = 0$. Among the uncountable many distributions belonging to $\mathcal{H}^a$, two possible choices for nature $\pi$ on $s$ and $a$ are $\pi(s, a) = \{(t, \frac{2}{3}), (u, \frac{2}{3})\}$ and $\pi(s, a) = \{(t, \frac{1}{3}), (u, \frac{1}{3})\}$.  

\section{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 that 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 $\Phi$ is given by:

$$\Phi ::= Pr_{-\rho}[\Psi] \mid Pr_{\min \rightarrow}[\Psi] \mid Pr_{\max \rightarrow}[\Psi],$$

$$\Psi ::= a \mid \neg \Psi \mid \Psi \land \Psi \mid X\Psi \mid \Psi \cup \Psi,$$

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

Note that a PLTL formula $\Phi$ is just a probability operator on top of an LTL formula $\Psi$; this is clear by the semantics of $\Phi$ and $\Psi$: Given an IMDP $\mathcal{M}$ and a PLTL formula $Pr_{-\rho}[\Psi]$, we say that $\mathcal{M}$ satisfies $Pr_{-\rho}[\Psi]$, written $\mathcal{M} \models Pr_{-\rho}[\Psi]$, if $Pr^{\sigma,\pi}_{\mathcal{M}}((\xi \in IPaths \mid \xi \models \Psi)) \sim \rho$ for all $\sigma \in \Sigma$ and $\pi \in IPaths$, extending $Pr_{\min \rightarrow}[\Psi]$ and $Pr_{\max \rightarrow}[\Psi]$ for $\rho = 0$ and $\rho = 1$. 

\[ \pi \in \Pi, \text{ where } \xi \models \Psi \text{ is defined inductively as follows:} \]

\[
\begin{align*}
\xi \models a & \quad \text{if } a \in L(\xi[0]), \\
\xi \models \neg \Psi & \quad \text{if it is not the case that } \xi \models \Psi \text{ (also written } \xi \not\models \Psi), \\
\xi \models \Psi_1 \land \Psi_2 & \quad \text{if } \xi \models \Psi_1 \text{ and } \xi \models \Psi_2, \\
\xi \models X\Psi & \quad \text{if } \xi[1 \ldots] \models \Psi, \text{ and} \\
\xi \models \Psi_1 \cup \Psi_2 & \quad \text{if there is } n \in \mathbb{N} \text{ with } \xi[n \ldots] \models \Psi_2 \text{ and for each } 0 \leq i < n, \text{ then } \xi[i \ldots] \models \Psi_1.
\end{align*}
\]

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

\[
Pr_{\text{opt}=?}[\Psi] = \text{opt}_{\sigma \in \Sigma, \pi \in \Pi} \Pr_{M}^{\sigma, \pi}(\{ \xi \in \text{Paths} \mid \xi \models \Psi \}).
\]

### 3 MULTI-OBJECTIVE ROBUST CONTROL OF IMDPs

In this section, we start by considering two main classes of properties for IMDPs: 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/\(\omega\)-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 \([T]_{\leq k}^\leq p\) 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 \([T]_{\leq k}^\leq p\) by MDP \(M\) under strategy \(\sigma \in \Sigma\) is denoted by \(M|_{\sigma} \models_{\Pi} [T]_{\leq k}^\leq p\) and indicates that the probability of the set of all paths that reach \(T\) under \(\sigma\) satisfies the bound \(\sim p\) for every choice of nature \(\pi \in \Pi\). Formally, \(M|_{\sigma} \models_{\Pi} [T]_{\leq k}^\leq p \iff \Pr_{M}(\Diamond \leq k T) \sim p\) where \(\Pr_{M}(\Diamond \leq k T) = \text{opt}_{\pi \in \Pi} \Pr_{M}^{\sigma, \pi}(\{ \xi \in \text{Paths} \mid \exists i \leq k; \xi[i] \in T \})\) and \(\text{opt} = \min\) if \(\sim = \geq\) and \(\text{opt} = \max\) if \(\sim = \leq\). Furthermore, \(\sigma\) is referred to as a robust strategy.

**Definition 6 (Reward Predicate & its Robust Satisfaction).** A reward predicate \([r]_{\leq k}^\leq r\) 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 \([r]_{\leq k}^\leq r\) by MDP \(M\) under strategy \(\sigma \in \Sigma\) is denoted by \(M|_{\sigma} \models_{\Pi} [r]_{\leq k}^\leq r\) and indicates that the expected total reward over the set of all paths under \(\sigma\) satisfies the bound \(\sim r\) for every choice of nature \(\pi \in \Pi\). Formally, \(M|_{\sigma} \models_{\Pi} [r]_{\leq k}^\leq r \iff \text{ExpTot}_{M}^{\sigma, k}[r] \sim r\) where \(\text{ExpTot}_{M}^{\sigma, k}[r] = \text{opt}_{\pi \in \Pi} \int_\xi r[k](\xi) \Pr_{M}^{\sigma, \pi}\) and \(\text{opt} = \min\) if \(\sim = \geq\) and \(\text{opt} = \max\) if \(\sim = \leq\). Furthermore, \(\sigma\) 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 \(w \in \mathbb{R}^n\), a vector of time bounds \(k = (k_1, \ldots, k_n) \in (\mathbb{N} \cup \{\infty\})^n\) and reward structures \(r = (r_1, \ldots, r_n)\) for an MDP \(M\), the weighted reward sum \(w \cdot r[k]\) over a path \(\xi\) is defined as

\[ w \cdot r[k](\xi) = \sum_{i=1}^{n} w_i \cdot r_i[k](\xi). \]

The expected total weighted sum is defined as \(\text{ExpTot}_{M}^{\sigma, k}[w \cdot r] = \max_{\pi \in \Pi} \int_\xi w \cdot r[k](\xi) \Pr_{M}^{\sigma, \pi}\) for bounds \(\leq\) and accordingly minimises over natures for \(\geq\); for a given strategy \(\sigma\), we have:

\[ \text{ExpTot}_{M}^{\sigma, k}[w \cdot r] = \sum_{i=1}^{n} w_i \cdot \text{ExpTot}_{M}^{\sigma, k_i}[r_i]. \]
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 \( \varphi \) is satisfied by IMDP \( M \) under strategy \( \sigma \) for every choice of nature \( \pi \in \Pi \), denoted by \( M |_{\sigma} \models_{\Pi} \varphi \) if, for each \( 1 \leq i \leq n \), we have \( M |_{\sigma} \models_{\Pi} \varphi_i \). We refer to \( \sigma \) as a robust strategy. Furthermore, we call \( \varphi \) a basic multi-objective predicate if it is of the form \( ([r_1]_{\geq r_1}, \ldots, [r_n]_{\geq r_n}) \), 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 \( M \) and a multi-objective predicate \( \varphi \), the synthesis query asks if there exists a robust strategy \( \sigma \in \Sigma \) such that \( M |_{\sigma} \models_{\Pi} \varphi \).

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

The next type of query is multi-objective quantitative queries, which are defined as follows:

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

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

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

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

The last type of query 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 \( M \) and a multi-objective predicate \( \varphi \), a Pareto query is of the form \( \text{Pareto}([o_1]_{\min \leq k_1}, \ldots, [o_n]_{\min \leq k_n}) \), where each \( [o_i]_{\min \leq k_i} \) 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 \( \text{opt}_i \in \{\min, \max\} \). We define the set of achievable values as \( A = \{ x \in \mathbb{R}^n \mid ([o_1]_{\leq k_1}, \ldots, [o_n]_{\leq k_n}) \text{ is satisfiable} \} \) where \( \sim_i = \leq \) if \( \text{opt}_i = \min \), or \( \sim_i = \leq \) if \( \text{opt}_i = \max \). Then,

\[
\text{Pareto}([o_1]_{\min \leq k_1}, \ldots, [o_n]_{\min \leq k_n}) = \{ x \in A \mid x \text{ is Pareto optimal} \}.
\]

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 \( M \) and a synthesis query \( \varphi \) are given. Let \( \varphi = ([T_1]_{\leq k_1}, \ldots, [T_n]_{\leq k_n}, [r_{n+1}]_{\leq k_{n+1}}, \ldots, [r_m]_{\leq k_m}) \). We say that \( \varphi \) is reward-finite if for each \( n + 1 \leq i \leq m \) such that \( k_i = \infty \), we have \( \sup_{\sigma \in \Sigma} \{ \text{ExpTo}_{M|^{|\sigma|}}^{\sigma, k_i} [r_i] \mid M |_{\sigma} \models_{\Pi} ([T_1]_{\leq k_1}, \ldots, [T_n]_{\leq k_n}) \} < \infty \).
In the next section, we provide a method to check for reward-finiteness assumption of a given \( IMDP \) \( M \) and a synthesis query \( \phi \), a preprocessing procedure that removes actions with non-zero rewards from the end components of \( M \), and a proof for the correctness of this procedure with respect to \( \phi \). In the rest of the article, we assume that all queries are reward-finite. Furthermore, for the soundness of our analysis, we also require that for any \( IMDP \) \( M \) and \( \phi \) given as in Assumption 1, the following properties hold: (i) each reward structure \( r_i \) assigns only non-negative values; (ii) \( \phi \) 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 \( \geq \).

### 3.2 A Procedure to Check Assumption 1

In this section, we discuss in detail how reward-finiteness assumption for a given \( IMDP \) \( M \) and a synthesis query \( \phi \) can be checked. Once it is known that the assumption is satisfied, the \( IMDP \) \( M \) 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 \( \phi \). To describe the procedure that checks Assumption 1, first, we need to define a counterpart of end components of \( MDPs \) for \( IMDPs \), 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 \( IMDP \) \( M \) is \( E_M = (S', \mathcal{A}') \), where \( S' \subseteq S \) and \( \mathcal{A}' \subseteq \bigcup_{s \in S'} \{ s \} \times \mathcal{A}(s) \) such that (i) \( \sum_{s' \in S'} b_{s'a}^{s'} = 1 \) for each \( s \in S' \), \( (s, a) \in \mathcal{A}' \), and \( b_{s'a}^{s'} \in \mathcal{H}_s \); 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 \( 0 \leq i \leq n - 1 \), we have \( \xi[i] \in S' \) and \( (\xi[i], \xi(i)) \in \mathcal{A}' \).

**Remark 13.** The SECs of an \( IMDP \) \( M \) 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 \( M \) are strictly greater than zero for the transitions whose upper probability bounds are non-zero, the underlying graph structure of \( M \) is identical to the graph structure of every \( MDP \) it contains. Therefore, a SEC of \( M \) 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

\[
\sup_{\sigma \in \Sigma} \inf_{\pi \in \Pi} \text{occ}_{\pi}(s, a) < \infty,
\]

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

**Proof.** If \( (s, a) \) is not contained in a SEC of \( M \), 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. \( \square \)

**Proposition 15.** Let \( E_M = (S', \mathcal{A}') \) denote a SEC of \( IMDP \) \( M \). Then, we have

\[
\text{sup}_{\sigma \in \Sigma} \{ \text{ExpTot}_{E_M}^{\sigma \infty}[r] | M|_{\sigma} \models [\mathcal{P}[1]_{\xi[k]} \cdots \mathcal{P}[n]_{\xi[n]}] \} = \infty \text{ for a reward structure } r \text{ of } M \text{ if and only if there exists a strategy } \sigma \text{ of } M \text{ that } M|_{\sigma} \models [\mathcal{P}[1]_{\xi[k]} \cdots \mathcal{P}[n]_{\xi[n]}], E_M \text{ is reachable under } \sigma, \text{ and } r(\xi[i], \xi(i)) > 0, \text{ where } \xi \text{ is a path under } \sigma \text{ with } \xi[i] \in S' \text{ and } (\xi[i], \xi(i)) \in \mathcal{A}'(\xi[i]) \text{ for some } i \geq 0.
\]

We can now construct, from \( M \), an \( IMDP \) \( \tilde{M} \) that is equivalent to \( M \) in terms of satisfaction of \( \phi \) 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 \( MDPs \) and is as follows: First, remove action \( a \) from \( \mathcal{A}(s) \)
if \( (s, a) \) is contained in a SEC and \( r(s, a) > 0 \) 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 \( \sigma \) of \( M \) such that \( \text{ExpTot}^{\sigma, \infty}_M[r] = x < \infty \) and \( M|_{\sigma} \models \varphi \) if and only if there is a strategy \( \overline{\sigma} \) of \( \overline{M} \) such that \( \text{ExpTot}^{\overline{\sigma}, \infty}_{\overline{M}}[r] = x \) and \( M|_{\varphi} \models \overline{\varphi} \).

### 3.3 Multi-objective Robust Strategy Synthesis

We first study the computational complexity of multi-objective robust strategy synthesis problem for \( IMDPs \). Formally,

**Theorem 17.** Given an \( IMDP \ M \) and a multi-objective predicate \( \varphi \), the problem of synthesising a strategy \( \sigma \in \Sigma \) such that \( M|_{\sigma} \models \varphi \) is \( \text{PSPACE} \)-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 \( IMDP \ M = (S, \bar{s}, \mathcal{A}, I) \) and a multi-objective predicate \( \varphi = ([T_1]_{s_1^{k_1}}, \ldots, [T_n]_{s_n^{k_n}}, [r_{n+1}]_{s_{n+1}^{r_{n+1}}}, \ldots, [r_m]_{s_m^{r_m}}) \), let \( M' = (S', \bar{s}', \mathcal{A}', I') \) be the \( IMDP \) whose components are defined as follows:

- \( S' = S \times 2^{[1, \ldots, n]} \),
- \( \bar{s}' = (\bar{s}, \emptyset) \),
- \( \mathcal{A}' = \mathcal{A} \times 2^{[1, \ldots, n]} \) and
- for all \( s, s' \in S, a \in \mathcal{A}, \) and \( v, v', v'' \subseteq \{1, \ldots, n\} \),

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

Now, let \( \varphi' = ([r_{T_1}]_{s_1^{p_1}}, \ldots, [r_{T_n}]_{s_n^{p_n}}, [r_{n+1}]_{s_{n+1}^{r_{n+1}}}, \ldots, [r_m]_{s_m^{r_m}}) \) where, for each \( i \in \{1, \ldots, n\} \),

\[
p'_i = \begin{cases} 
p_i & \text{if } \sim_i = \geq, \\
-p_i & \text{if } \sim_i = \leq,
\end{cases}
\]

and, for each \( j \in \{n + 1, \ldots, m\} \),

\[
r'_j = \begin{cases} 
r_j & \text{if } \sim_j = \geq, \\
-r_j & \text{if } \sim_j = \leq,
\end{cases}
\]

Then \( \varphi \) is satisfiable in \( M \) if and only if \( \varphi' \) is satisfiable in \( M' \).

Intuitively, the transformation of \( M \) to \( M' \) 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 \( v \subseteq \{1, \ldots, n\} \) 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 \([T_I]_{\leq p_i}^{\leq k_i} \) to \([T_I]_{\geq -p_i}^{\leq k_i+1} \) and \([r_j]_{\leq -r_j}^{\leq k_j} \) to \([-r_j]_{\geq -r_j}^{\leq k_j} \).

In Forejt et al. [2012], Proposition 2, the thresholds are not made negative, and this is a flaw:
Consider, for instance, the \(IMDP \) \(M\), 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 \(\varphi = ([s_1])_{\leq 0.3} \). Clearly, \(M|_{s_0} \models \varphi\), by \(s_0\) being the state choosing \(a\) in \(s_0\). In the transformed \(IMDP \) \(M'\), the newly added reward structure \(r_{\{s_1\}}\) assigns reward 0 to \(((s_0, \emptyset), (a, \emptyset))\) and reward −1 to \(((s_0, \emptyset), (b, \{1\}))\); \(\varphi\) is transformed to \(\varphi' = [r_{\{s_1\}}]_{\leq 0.5}^{\leq 0.5}\), which is still satisfiable by the strategy choosing \((a, \emptyset)\) in \((s_0, \emptyset)\). Since \(M\) is also an \(MDP\), we can apply the transformation given in Forejt et al. [2012], Proposition 2: \(M'\) and \(r_{\{s_1\}}\) are the same while \(\varphi\) is transformed to \(\psi = [r_{\{s_1\}}]_{\leq 0.5}^{\leq 0.5}\) (instead of \([r_{\{s_1\}}]_{\leq 0.5}^{\geq 0.5}\)), which is obviously unsatisfiable given that \(r_{\{s_1\}}\) assigns only non-positive values to each state-action pair.

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

In Figure 2(b), 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 \(\frac{1}{2}\) to reach \(T\). Afterwards, the reward obtainable
linearly decreases. If we require a reachability probability for \(T\) of \(\frac{5}{6}\), then 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 \(\frac{1}{2}\), 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 \(\frac{2}{3}\).

By means of Proposition 18, for robust strategy synthesis, we therefore need to only consider
the basic multi-objective predicates of the form \(([r_1]_{\geq r_1}, \ldots, [r_n]_{\geq r_n})\). For such a predicate, we define its Pareto curve as follows:

---

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

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 $\varepsilon$-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 to solve the multi-objective predicate, which is essentially based on normalising multiple 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 $\mathcal{P}_{M,\varphi}$ 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 $w$ are generated and corresponding to each of them, a $w$-weighted sum of $n$ objectives is optimised through lines 8–9. The optimal strategy $\sigma$ is then used to generate a point $g$ on the Pareto curve $\mathcal{P}_{M,\varphi}$. We collect all these points in the set $X$. The multi-objective predicate $\varphi$ is satisfiable once we realise that $r$ 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 $g_1, \ldots, g_t$ on the Pareto curve $\mathcal{P}_{M,\varphi}$ are generated, each of which corresponds to a deterministic strategy $\sigma_{g_i}$ for the current point $g_i$. The resulting optimal strategy $\sigma_{opt}$ 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 MDPs. With respect to the competitive semantics needed here, one can instead transform MDPs to 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-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 MDP so the core part, i.e., lines 8–9 of Algorithm 1, can be solved with time complexity polynomial in |M|.

Algorithm 2 represents a value iteration–based algorithm that extends the value iteration–based algorithm of Forejt et al. [2012] and adjusts it for MDP models by encoding the notion of

\begin{algorithm}
\noindent Input: An MDP M, weight vector w, reward structures r = (r_1, \ldots, r_n), time-bound vector k ∈ (N ∪ {∞})^n, threshold ε.
\noindent Output: strategy σ maximising ExpTot^σ_M [w · r], g := (ExpTot^σ_M [r_i] \mid 1 ≤ i ≤ n)
\begin{algorithmic}
  \STATE x := 0; x^1 := 0; \ldots; x^n := 0;
  \STATE y := 0; y^1 := 0; \ldots; y^n := 0;
  \STATE σ^∞(s) := ⊥ for all s ∈ S;
  \WHILE{δ > ε}
    \FOR{foreach s ∈ S do}
      \STATE \sigma^∞(s) := \arg \max_{a \in A(s)} \left( \sum_{i \mid |k_i| > 0} w_i \cdot r_i(s, a) + \min_{b_i \in H_i} b_i^a(s') \cdot x_{s'} \right);
      \STATE \delta := \max_{s \in S} \left( y_s - x_s \right);
      \STATE x := y; x^1 := 0; \ldots; x^n := 0;
    \ENDFOR
  \ENDWHILE
  \FOR{foreach s ∈ S do}
    \STATE y^i := r_i(s, 0^∞(s)) + \sum_{s' \in S} b^0_σ(s') \cdot x_{s'};
    \STATE δ := \max_{s \in S} \left( y^i_s - x^i_s \right);
    \STATE x^1 := 0; \ldots; x^n := 0;
  \ENDFOR
  \FOR{j := \max\{|k_p| < \infty \mid p \in [1, \ldots, n]\} down to 1 do}
    \STATE y^i_s := \max_{a \in A(s)} \left( \sum_{i \mid |k_i| > j} w_i \cdot r_i(s, a) + \min_{b_i \in H_i} b_i^a(s') \cdot x_{s'} \right);
    \STATE δ^j := \max_{s \in S} \left( y^i_s - x^i_s \right);
    \STATE x := y; x^1 := 0; \ldots; x^n := 0;
  \ENDFOR
  \FOR{i := 1 to n do}
    \STATE δ := δ^i;
    \STATE σ acts as σ^i in jth step when j < max_{i \in [1, \ldots, n]} |k_i| and as σ^∞ afterwards;
  \ENDFOR
\RETURN σ, g;
\end{algorithmic}
\end{algorithm}
Algorithm 3: Algorithm for solving robust quantitative queries

Input: An IMDP \( M \), objective \( [r_1]_{k_1}^{\leq} \), multi-objective predicate \( ([r_2]_{k_2}^{\geq}, \ldots, [r_n]_{k_n}^{\geq}) \)

Output: value of \( qnt([r_1]_{k_1}^{\leq}, ([r_2]_{k_2}^{\geq}, \ldots, [r_n]_{k_n}^{\geq}) \)

begin
1. \( X = \emptyset; \)
2. \( r = (r_1, \ldots, r_n); \)
3. \( k = (k_1, \ldots, k_n); \)
4. \( r = (\min_{\sigma \in \Sigma} \text{ExpTot}^k_M[r_1], r_2, \ldots, r_n); \)
5. while \( r \notin X_\downarrow \) or \( w \cdot g > w \cdot r \) do
6.   Find \( w \) separating \( r \) from \( X_\downarrow \) such that \( w_1 > 0; \)
7.   Find strategy \( \sigma \) maximising \( \text{ExpTot}^\sigma_M[w \cdot r]; \)
8.   \( g := (\text{ExpTot}^\sigma_M[r_1])_1 \leq i \leq n; \)
9.   if \( w \cdot g < w \cdot r \) then
10.     return \( \perp; \)
11.   \( X = X \cup \{g\}; \)
12.   \( r_1 := \max\{r_1, \max\{r' \mid (r', r_2, \ldots, r_n) \in X_\downarrow\}\}; \)
13. return \( r_1; \)

robustness. More precisely, the core difference is at lines 7 and 19, where the optimal strategy is computed to be robust against any choice of nature.

Theorem 22. Algorithm 1 is sound, complete, and has runtime exponential in \( |M|, k, \) and \( n. \)

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 that minimizes/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.

To present the algorithm, consider the quantitative query \( qnt([r_1]_{k_1}^{\leq}, ([r_2]_{k_2}^{\geq}, \ldots, [r_n]_{k_n}^{\geq}) \). Algorithm 3, similarly to Algorithm 1, generates a sequence of points \( g \) on the Pareto curve from a sequence of weight vectors \( w \). To optimise the objective \( r_1 \), a sequence of lower bounds \( r_1 \) is generated that 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 MDPs [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 $\varepsilon$-approximation of the Pareto curve.

3.6 PLTL and $\omega$-regular Properties

PLTL formulas, or in general $\omega$-regular properties, allow one to express properties of an MDP with respect to its infinite behaviour. Examples of PLTL formulas are: with probability at least 0.95, the MDP will never be trapped in an error state ($\Pr_{\geq 0.95}[G(F = \mathit{error})]$); almost surely, whenever a request arrives, eventually a response is provided ($\Pr_{\geq 1}[G(\mathit{req} \Rightarrow \mathit{resp})]$); with probability at least

0.99, the system eventually becomes stable \((Pr_{\geq 0.99}[FGstable])\). The classical approach to verify a PLTL formula \(Pr_{\leq p}[\Psi]\), or an \(\omega\)-regular property, against an MDP \(M\) consists in constructing a deterministic Rabin automaton (DRA) \(R_{\Psi}\) accepting the same words satisfying \(\Psi\), then construct the product \(M \times R_{\Psi}\), find the accepting maximal end components of \(M \times R_{\Psi}\), 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 \(\omega\)-regular properties against an IMDP \(M\). In practice, the construction is the extension to IMDPs of the approach for MDPs.

**Definition 24 (Product IMDP \(M \times R\)).** For given IMDP \(M = (S, \bar{s}, A, 1, AP, L)\) and DRA \(R = (Q, q_0, 2AP, T, Acc)\) with \(Acc = \{(A_1, R_1), \ldots, (A_k, R_k)\}\), the product \(M \times R\) is the IMDP \(M \times R = (S \times Q, s', A, 1', Q, L')\) where

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

Similarly to the MDP case, we can prove that the probability of \(M\) to satisfy \(\Psi\) equals the probability of reaching accepting SECs in \(M \times R_{\Psi}\), where a SEC \(M'\) of \(M \times R_{\Psi}\) 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 \(M\) be an IMDP, \(\Psi\) an LTL formula, and \(U\) be the union of all accepting SECs in \(M \times R_{\Psi}\). Then for each strategy \(\sigma\) for \(M\) there exists a strategy \(\sigma'\) for \(M \times R_{\Psi}\) such that for each nature \(\pi\) for \(M \times R_{\Psi}\) there exists a nature \(\pi'\) for \(M \times R_{\Psi}\) such that

\[
Pr_{\Pi}^\sigma[M \times R_{\Psi}] \{ [\xi \in IPaths_M \mid \xi \models \Psi] \} = Pr_{\Pi}^{\sigma', \pi'}[M \times R_{\Psi}] \{ [\xi \in IPaths_{M \times R_{\Psi}} \mid \exists j \in \mathbb{N}: \xi[j] \in U] \}
\]

and vice versa.

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

As an immediate consequence of Theorem 25, we also have that the robust probability of satisfying \(\Psi\) under a strategy \(\sigma\) for \(M\) coincides with the robust probability of reaching accepting SECs under some strategy \(\sigma'\) for \(M \times R_{\Psi}\).

**Corollary 26.** Let \(M\) be an IMDP, \(Pr_{\leq p}[\Psi]\) a PLTL formula, and \(U\) be the union of all accepting SECs in \(M \times R_{\Psi}\); let \(\Pi'\) denote the set of natures for \(M \times R_{\Psi}\). Then for each strategy \(\sigma\) for \(M\) there exists a strategy \(\sigma'\) for \(M \times R_{\Psi}\) such that

\[
\text{opt} \frac{Pr_{\Pi}^\sigma[M \times R_{\Psi}] \{ [\xi \in IPaths_M \mid \xi \models \Psi] \}}{\Pi} = \text{opt} \frac{Pr_{\Pi'}^{\sigma', \pi'}}{\Pi'}[M \times R_{\Psi}] \{ [\xi \in IPaths_{M \times R_{\Psi}} \mid \exists j \in \mathbb{N}: \xi[j] \in U] \}
\]

and vice versa, where \(\text{opt} = \min\) if \(\sim = \leq\) and \(\text{opt} = \max\) if \(\sim = \geq\).

By means of Theorem 25 and Corollary 26, we can extend the results about multi-objective (quantitative) queries (cf. Sections 3.1 and 3.4) and Pareto queries (cf. Section 3.5) to general PLTL and \(\omega\)-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 \(M = (S, \bar{s}, A, 1)\) and a basic multi-objective predicate \([|r_1| \leq k_1, \ldots, |r_n| \leq k_n]\). For clarity, we assume that all \(k_i = \infty\); we discuss the extension to \(k_i < \infty\)
Fig. 3. Computing randomised strategies.

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 \( X = \{g_1, \ldots, g_m\} \) of reward vectors \( g_i = (g_{i,1}, \ldots, g_{i,n}) \) and their corresponding set of strategies \( \Sigma = \{\sigma_1, \ldots, \sigma_m\} \), where strategy \( \sigma_i \) achieves the reward vector \( g_i \).

In the descriptions of the given algorithms, the strategies \( \sigma_i \) 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 \( \sigma_i : S \rightarrow A \) or, equivalently, as functions \( \sigma_i : S \times A \rightarrow \{0, 1\} \) 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 \( P = \{p_1, \ldots, p_m\} \) of the probabilities with which each of these strategies shall be executed. If we execute each \( \sigma_i \) with its according probability \( p_i \), then the vector of total expected rewards is \( g = \sum_{i=1}^{m} p_i \cdot g_i \). Let \( r = (r_1, \ldots, r_n) \) 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 fulfill the constraints \( \sum_{i=1}^{m} p_i \cdot g_i \geq r, \sum_{i=1}^{m} p_i = 1 \), 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 \( g \), we initially randomly choose one of the memoryless deterministic strategies \( \sigma_i \) according to their probabilities in \( P \). Afterwards, we just keep executing the chosen \( \sigma_i \). 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 \( \sigma_i \).

This process of obtaining the expected rewards \( g \) indeed uses memory, because we have to remember the deterministic strategy that 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 \( \sigma_i \) with probability \( p_i \) in each step anew.

Example 27. Consider the \( \text{IMDP} \) in Figure 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 \( \pi \). There is only a single reward structure, indicated by the underlined numbers. If we choose \( a \) in state \( s \), then 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.

We consider the strategies \( \sigma_a \), which chooses \( a \) in each state, and \( \sigma_b \), which chooses \( b \) in each state. With both strategies, we accumulate a reward of exactly 1. Therefore, if we choose to execute \( \sigma_a \) with probability 0.5 and \( \sigma_b \) with the same probability, this process will lead to a reward of 1 as well.
Now, consider a strategy that chooses the action selected by $\sigma_a$ in each state with probability 0.5, and with another probability chooses the action selected by $\sigma_b$. 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.

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 $\pi: \text{FPaths} \times \mathcal{A} \to \text{Disc}(S)$; that is, $\pi: S \times \mathcal{A} \to \text{Disc}(S)$. The particular choice of $\pi$ is not important, which is due to the fact that our algorithms are robust against any choice of nature. We then let $\chi^\sigma_i(s)$ denote the probability to be in state $s$ at step $i$ when strategy $\sigma$ is used (using nature $\pi$ and under the condition that we have started in $s$).

For any $\sigma \in \Sigma$, we have $\chi^\sigma_i(s) = \sum_{\xi \in \text{FPaths}(s)} \chi^\sigma_i(s) \cdot \pi_{\text{FPaths}(s)}(\xi) \cdot \chi^\sigma_{i+1}(s')$, which can be shown to be equivalent to the inductive form $\chi^\sigma_i(s) = 1$ and $\chi^\sigma_i(s) = 0$ for $s \neq s$, and $\chi^\sigma_{i+1}(s) = \sum_{s' \in S} \pi(s', \sigma(s')) \cdot \chi^\sigma_i(s')$.

The state-action frequency $y^\sigma(s, a)$ is the number of times action $a$ is chosen in state $s$ when using strategy $\sigma$. We then have that $y^\sigma(s, a) = \sum_{i=0}^\infty \chi^\sigma_i(s) \cdot \sigma(s, a)$. 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, $y(s, a) = \sum_{i=1}^\infty p_i \cdot y^\sigma_i(s, a)$ for all $s, a$. To construct a memoryless randomised strategy $\sigma$, 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 Figure 3, we have $y^\sigma_a(s, a) = 1$, $y^\sigma_a(s, b) = 0$, $y^\sigma_a(u, a) = 0$, $y^\sigma_a(u, b) = 0$, $y^\sigma_b(s, a) = 1$, $y^\sigma_b(s, b) = 0$, $y^\sigma_b(u, a) = 0$, and $y^\sigma_b(u, b) = 0$. If we choose both $\sigma_a$ and $\sigma_b$ with probability 0.5, then 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 $\sigma$ 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.

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 $\sigma_i: S \times \{0, \ldots, k_{\max}\} \to \mathcal{A}$ or equivalently $\sigma_i: S \times \{0, \ldots, k_{\max}\} \times \mathcal{A} \to \{0, 1\}$ where $\sigma_i(s, j, a) = 1$ if $\bar{\sigma}(s, j) = a$ and $\sigma_i(s, j, \cdot) = 0$ otherwise. For step $i$ with $i < k_{\max}$, a strategy $\sigma$ 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 $\pi: S \times \mathcal{A} \times \{0, \ldots, k_{\max}\} \to \text{Disc}(S)$. The computation of the randomised strategy changes accordingly: For any $\sigma \in \Sigma$, we have $x^\sigma_0(s) = 1$, $x^\sigma_0(s) = 0$ for $s \neq s$, and $x^\sigma_i(s) = \sum_{s' \in S} \pi(s', \sigma(s'), i') \cdot y^\sigma_i(s')$ where $i' = \min\{i, k_{\max}\}$. Also, the state-action frequencies are now defined as step-dependent. For $i \in \{0, \ldots, k_{\max} - 1\}$, we define $y^\sigma(s, i, a) = x^\sigma_i(s) \cdot \sigma(s, i, a)$ and $y^\sigma(s, k_{\max}, a) = \sum_{i \geq k_{\max}} x^\sigma_i(s) \cdot \sigma(s, a)$.

The mixed state-action frequency is then $y(s, i, a) = \sum_{i=1}^m p_i \cdot y^\sigma(s, i, a)$. 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 $\sigma(s, i, \cdot)$, while for $i \geq k_{\max}$, we use decisions from $\sigma(s, k_{\max}, \cdot)$.

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 $S_{unrolled} = \bigcup_{i \in \{0, \ldots, k_{\max}\}} S_i$. We call each set of states $S_i$ a layer. For each state $s \in S$ and $i \in \{0, \ldots, k_{\max}\}$, we have $s_i \in S_i$. If we have a transition from a state $s$ to a state $s'$, then in the unrolled $IMDP$ for all $i \in \{0, \ldots, k_{\max} - 1\}$, 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 $r_{unrolled}(s_i, a, s_{i+1}') = I(s, a, s')$ for some states $s, s'$ and some action $a$ and zero else, and then $r_{unrolled}(s_{k_{\max}}, a, s'_{k_{\max}}) = I(s, a, s')$. Thus, there are only transitions from 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 $r_{unrolled}(s_i, a) = r(s, a)$ 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, $r_{unrolled}(s_i, a) = r(s, a)$ for $i \in \{0, \ldots, k - 1\}$. However, to simulate the step bound, we let $r(s_i, a) = 0$ 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, $\sigma(s, i, a) = \sigma_{unrolled}(s_i, a)$. 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., $y(s, i, a) = y_{unrolled}(s_i, a)$. Note that for $i < k_{\max}$ in $y_{unrolled}, \sigma(s_i, a) = \sum_{j=0}^{\infty} x_j^\sigma(s_i) \cdot \sigma(s_i, a)$ only the summand for $j = i$ is relevant. This is the case, because by construction of the unrolled $IMDP$ for the other $j$ with $j \neq i$, we have $x_j^\sigma(s_i) = 0$. Thus, $y_{unrolled, \sigma}(s_i, a) = x_i^\sigma(s_i) \cdot \sigma(s_i, a)$. Accordingly, for $y_{unrolled, \sigma}(s_{k_{\max}}, a) = \sum_{j=k_{\max}}^{\infty} x_j^\sigma(s_{k_{\max}}) \cdot \sigma(s_{k_{\max}}, a)$ only $j$ with $j \geq k_{\max}$ are relevant and thus $y_{unrolled, \sigma}(s_{k_{\max}}, a) = \sum_{j=k_{\max}}^{\infty} x_j^\sigma(s_{k_{\max}}) \cdot \sigma(s_{k_{\max}}, a)$.

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] and 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 Figure 4(a). 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 $IMDP M$ and then computing strategies on $M$ as in Luna et al. [2014a, 2014b, 2014c].
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 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: $([r_p]_{\leq \infty} \geq 0.9, [r_d]_{\leq \infty} \leq 50)$. We first computed the Pareto curve for the property, which is shown in Figure 4(b), 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

$$\varphi_1 = ([r_p]_{\leq \infty} \geq 0.95, [r_d]_{\leq \infty} \leq 50), \quad \varphi_2 = ([r_p]_{\leq \infty} \geq 0.90, [r_d]_{\leq \infty} \leq 45), \quad \varphi_3 = ([r_p]_{\leq \infty} \geq 0.66, [r_d]_{\leq \infty} \leq 25).$$

We then simulated the robot under each strategy 500 times. The statistical results of these simulations are consistent with the bounds in $\varphi_1$, $\varphi_2$, and $\varphi_3$. The collision-free robot trajectories are
Interval MDPs with Multiple Objectives: From Robust Strategies to Pareto Curves

Fig. 5. Robot sample paths under strategies for $\varphi_1$, $\varphi_2$, and $\varphi_3$.

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 $(P_{\text{max}} \geq \psi_i, [r_t]_{\text{min}})\,$ for $i \in \{4, 5\}$.

shown in Figure 5. These trajectories illustrate that the robot is conservative under $\varphi_1$ and takes a longer route with open spaces around it to reach the target to be safe (Figure 5(a)), while it becomes reckless under $\varphi_3$ and tries to go through a narrow passage with the knowledge that its motion is noisy and could collide with the obstacles (Figure 5(c)). This risky behaviour, however, is required to meet the bound on the expected travelled distance in $\varphi_3$. The sample trajectories for $\varphi_2$ (Figure 5(b)) demonstrate the stochastic nature of the strategy. That is, the robot probabilistically chooses between being safe and reckless to satisfy the bounds in $\varphi_2$.

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 Section 5.1, and the warehouse map is shown in Figure 6(a). 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 Section 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,
Fig. 7. Robot sample paths under strategies for $\varphi_6$–$\varphi_{11}$. 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-$p_2$-$p_3$-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-$D$.

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 $\text{IMDP}$ 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:

$$\psi_4 = G\neg\text{obstacle} \land F(p_1 \land \text{Fdrop-off}), \quad \psi_5 = G\neg\text{obstacle} \land \bigwedge_{i=1}^{3} (\neg\text{drop-off} \cup p_i) \land \text{Fdrop-off}.$$ 

Therefore, the pair of objectives for each task can be expressed as $(P_{\text{max}}[\psi_i], [r_T]_{\text{min}}^\infty)$ 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 $\text{IMDP}$s for tasks $\psi_4$ and $\psi_5$. The $\text{IMDP}$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 Figure 6(b). Then, we computed the robust strategies for the following properties (Pareto points):

$$\varphi_6 = (Pr_{\geq 0.43}[\psi_4], [r_T]_{\leq 90}), \quad \varphi_7 = (Pr_{\geq 0.67}[\psi_4], [r_T]_{\leq 200}), \quad \varphi_8 = (Pr_{\geq 0.80}[\psi_4], [r_T]_{\leq 270}),$$

$$\varphi_9 = (Pr_{\geq 0.41}[\psi_5], [r_T]_{\leq 130}), \quad \varphi_{10} = (Pr_{\geq 0.49}[\psi_5], [r_T]_{\leq 200}), \quad \varphi_{11} = (Pr_{\geq 0.65}[\psi_5], [r_T]_{\leq 400}).$$

The sample robot trajectories under these strategies are shown in Figure 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 $\varphi_6$–$\varphi_8$ that correspond to task $\psi_4$, 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 (Figure 7(a)).
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 Figure 7(b). 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 (Figure 7(c)). A similar trend is observed for \( \psi_6 - \psi_{11} \) but at larger time duration, since task \( \psi_5 \) requires a collection of three products, as shown in Figures 7(d)–7(f). 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 \( \psi_6 - \psi_{11} \), validating the proposed approach.

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] and 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 visitors can choose what they would like to visit and pay tickets according to their preferences. 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 art to a visitor may depend on many factors, such as price, weather, the length of the queue at that moment, and so on, 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 southwest corner of the museum and \((n - 1, n - 1)\) the northeast corner. 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 northeast, southeast, northwest, and southwest 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 northeast 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 Figure 8(a). 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 receives 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 \( r_j \) being the reward structure for the number
of steps and $r_p$ denoting the penalty accumulated, \(([r_x] \leq 40, [r_p] \leq 70)\) requires that we leave the museum within 40 steps but with a penalty of no more than 70. The red arrows indicate a strategy that 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 Figure 8(b). 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 Figure 9, we provide strategies for different points on the Pareto curve in Figure 8(a). 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 Figure 9(a). 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 Figure 9(b) and Figure 9(c), we give the tourist somewhat more time—31 steps—so 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 Figure 9(c) 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 Figure 9(d) and Figure 9(e), then 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 Figure 9(f), which circumvents all closed exhibitions.

6 CONCLUDING REMARKS

In this article, 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 IMDPs 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 IMDPs with multiple objectives can also be applied to approximate Pareto curves for MDPs with convex uncertainty sets as well as \( \omega \)-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 IMDPs, which is left open in this article.

APPENDIX

A PROOFS OF THE RESULTS ENUNCIATED IN THE ARTICLE

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

To prove Theorem 17, we need to define the multiple reachability problem for MDPs. Formally,

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

The proof also makes use of the following lemma:

Lemma 30 (Complexity of the Multi-objective Reachability Problem for MDPs [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 \( M \) generated from \( M \) by replacing all probability values with point intervals. Since the multiple reachability problem for MDPs is PSPACE-complete and the reduction is performed in polynomial time, solving the robust strategy synthesis problem for MDPs 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 \( g \) on a unique face of the Pareto curve is identified. The number of faces of the Pareto curve \( P_{M, \varphi} \) is, in the worst case, exponential in \( |M|, k, \) 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 \( |M| \) 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 \( \text{ExpTot}_{M}^{\sigma, k}[w \cdot r] \) across the range of strategies \( \sigma \in \Sigma \) can be
formulated as the following optimisation problem:

$$\min_{x} \; x^T 1$$

subject to:

$$x_s \geq \sum_{i=1}^{n} w_i \cdot r_i(s, a) + \min_{b_s^a \in H^a_s} x^T b_s^a \quad \forall s \in S, \forall a \in A(s).$$

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:

$$\max_{x} \; -x^T 1$$

subject to:

$$x_s \geq \sum_{i=1}^{n} w_i \cdot r_i(s, a) + \min_{b_s^a \in H^a_s} x^T b_s^a \quad \forall s \in S, \forall a \in A(s).$$

As it is clear from the set of constraints in the latter optimization problem, the inner optimisation problem with fixed $$\tilde{b}$$, $$1^T b_s^a = 1$$, we can rewrite the latter inner optimisation problem as:

$$P(x) := \min_{b_s^a \in H^a_s} x^T b_s^a.$$ 

Based on the general description of the interval uncertainty set $$H^a_s = \{ b_s^a | 0 \leq b_s^a \leq b_s^a \leq \tilde{b}_s \leq 1, 1^T b_s^a = 1 \}$$, we can rewrite the latter inner optimisation problem as:

$$P(x) := \min_{b_s^a \in H^a_s} x^T b_s^a$$

subject to:

$$1^T b_s^a = 1$$

$$b_s^a \leq b_s^a \leq \tilde{b}_s.$$ 

The dual of the above problem is formulated as follows:

$$D(x) := \max_{y_{j,1} \in H^a_j, y_{j,2}, y_{j,3}} \sum_{i=1}^{n} r_i(s, a) + y_{j,1}^{s,a} + b_s^a x^T y_{j,3}^s - b_s^a y_{j,2}$$

subject to:

$$x - y_{j,2} + y_{j,3} - y_{j,1} = 0$$

$$y_{j,2} \succeq 0, y_{j,3} \succeq 0.$$ 

Since the latter inner optimisation problem with fixed $$x$$ is an LP, therefore due to the strong duality theorem [Bertsimas and Tsitsiklis 1997], we have $$P^*(x) = D^*(x)$$ where $$P^*(x)$$ and $$D^*(x)$$ 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 $$M$$ with respect to the reward structure $$w \cdot r$$ is formulated as the following LP, which can in turn be solved in polynomial time.

$$\max_{x, y} \; -x^T 1$$

subject to:

$$x_s \geq \sum_{i=1}^{n} w_i \cdot r_i(s, a) + y_{j,1}^{s,a} + b_s^a x^T y_{j,3} - b_s^a y_{j,2}$$

$$x - y_{j,2} + y_{j,3} - y_{j,1} = 0$$

$$y_{j,2} \succeq 0, y_{j,3} \succeq 0.$$ 

$$\forall s \in S, \forall a \in A(s).$$

$$\forall s \in S, \forall a \in A(s).$$
PROOF OF Proposition 18. Let $v = \{ i \in [1, \ldots, n] \mid s \in T_i \} \setminus v$. By definition of the transition probability function, it follows that the only successors $(s', v')$ that can be reached from $(s, v)$ must have $v' = v \cup v_c$; moreover, the action performed for such a transition must be of the form $(a, v_c)$. This means that the sets $v_c$ and $v'$ are uniquely determined by the current state $(s, v)$: let $v: S' \rightarrow \{ 1, \ldots, n \}$ be the function such that $v(s, v) = \{ i \in [1, \ldots, n] \mid s \in T_i \} \setminus v$ for each $(s, v) \in s'$, $\forall C: S' \times A \rightarrow A'$ be the function such that $v_C((s, v), a) = (a, v(s, v))$ for each $(s, v) \in s'$ and $a \in A$, and $v_S: S' \times S \rightarrow S'$ be the function such that $v_S((s, v), s') = (s, v \cup v(s, v))$ for each $(s, v) \in s'$ and $s' \in s$.

It is immediate to see that every path $\pi' \in M'$, $\pi' = (s_0, v_0)(a_0, v_0')(s_1, v_1)(a_1, v_1')(s_2, v_2) \ldots$, is actually of the form $\pi' = (s_0, v_0)(a_0, v_0')(s_1, v_1)(a_1, v_1')(s_2, v_2) \ldots$ where $(s_{j+1}, v_{j+1}) = v_S((s_j, v_j), a_j)$ and $(s_{j+1}, v_{j+1}) = v_S((s_j, v_j), a_j)$ and $(s_{j+1}, v_{j+1}) = v_S((s_j, v_j), a_j)$ and $(s_{j+1}, v_{j+1}) = v_S((s_j, v_j), a_j)$. This means that we can define a bijection $\# : \text{Paths} \rightarrow \text{Paths}'$ as follows: Given a path $\pi = s_0a_0s_1a_1s_2 \ldots$ of $M$, $\#(\pi')$ is defined as $\#(\pi') = s_0a_0s_1a_1s_2 \ldots$ of $M'$, $\#(\pi')$ is defined as $\#(\pi') = s_0a_0s_1a_1s_2 \ldots$ of $M'$. Moreover, since the sequence of sets $v_Sv_Sv_S \ldots$ is monotonic non-decreasing with respect to the subset inclusion partial order, we have that, for a given $i \in [1, \ldots, n]$, if $i \in v_N$ for some $N \in \mathbb{N}$, then there exists exactly one $l \in \mathbb{N}$ such that $i \notin v_l$ for each $l < l$ and $i \in v_l$ for each $l \geq l$, i.e., $s_l$ is the first time a state $s \in T_i$ occurs along $b(\pi')$. Therefore, it follows that $i \notin v(s_l, v_l)$ for each $l \in \mathbb{N} \setminus \{ l \}$. This implies that $r_{T_i}(\pi'[l], \pi'([l])) = 1$ if $l = 0$ or $r_{T_i}(\pi'[l], \pi'([l])) = -1$ if $l = \leq$ while $r_{T_i}(\pi'[j], \pi'([j])) = 0$ for each $j \in \mathbb{N} \setminus \{ l \}$, thus

$$r_{T_i}[k](\pi') = \begin{cases} 1 & \text{if } l < k \text{ and } i \leq, \\ 0 & \text{otherwise.} \\ -1 & \text{if } l < k \text{ and } i \leq, \\ 0 & \text{otherwise.} \\ 

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

Similarly, for each $h \in \{ n+1, \ldots, m \}$, we get that $r_{C_h}[k](\pi') = r_{C_h}[k](\pi)$ if $i = \geq$ and $r_{C_h}[k](\pi') = -r_{C_h}[k](\pi)$ if $i = \leq$.

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

Suppose that $\pi$ is satisfiable in $M$: By definition, it follows that there exists a strategy $\sigma$ of $M$ such that $M |_{\sigma} \models \pi \phi$; that is, $M |_{\sigma} \models \pi \left[ T_{1, k_1} \leq k \right] |_{\rho_i}$ for each $i \in [1, \ldots, n]$ and $M |_{\sigma} \models \pi \left[ T_{h, k_h} \leq k \right] |_{\rho_i}$ for each $h \in \{ n+1, \ldots, m \}$. Let $\sigma'$ be the strategy of $M'$ such that, for each finite path $\pi' \in P_{\text{Paths}}'$ and action $a \in A$, $\sigma(\pi')(v_A(\text{last}(\pi'), a)) = \sigma(\pi')(a)$, 0 otherwise. Intuitively, $\sigma'$ chooses the next action $(a, v)$ exactly as $\sigma$ chooses $a$, since $v$ is uniquely determined by $\pi'$. We claim that $\sigma'$ is such that $M' \models \pi \phi'$.

Let $i \in [1, \ldots, n]$ and consider $\pi'_i = [r_{T_i}|_{\geq k+1}]$: There are two cases depending on the original bound $\sim_i$.

If $i = \geq$, then $[r_{T_i}|_{\geq k+1}] = [r_{T_i}|_{\geq k+1}]$; $M' \models \pi \left[ T_{i, k_i} \geq k \right] |_{\rho_i}$ and if only if $\min_{i \in I_{\text{Paths}}'} \int_{\pi'} r_{T_i}[k_i + 1](\pi') \leq k \phi_M'$. Since for each path $\pi' \in P_{\text{Paths}}'$, $r_{T_i}[k_i + 1](\pi') = 1$ if there exists $k' \leq k - 1$ such that $b(\pi'[l]) = T_i$, $r_{T_i}[k_i + 1](\pi') = 0$ otherwise, by the way $I'$ and $\sigma'$ are defined, it follows that $\min_{i \in I_{\text{Paths}}'} \int_{\pi'} r_{T_i}[k_i + 1](\pi') \leq k \phi_M' = \min_{i \in I_{\text{Paths}}'} \int_{\pi'} r_{T_i}[k_i + 1](\pi') \leq k \phi_M'$.
\min_{\pi \in \Pi} \Pr_{M, \pi}^{\sigma, \pi} \{ \xi \in \mathcal{I} \mid \exists l \leq k: \xi[l] \in T_i \} \geq p_i, \text{ thus } \min_{\pi' \in \Pi} \int_{\xi} r_{T_i} [k_i + 1] (\xi') d\Pr_{M', \pi'}^{\sigma, \pi'} \geq p_i \text{ holds as well, hence } M' \mid_{\sigma'} \models_{\Pi} [r_{T_i}]_{\geq k_i + 1} = [r_{T_i}]_{\geq k_i + 1} \text{ is satisfied, as required.}

Consider now the second case: If \( \sim_i = \leq \), then \( [r_{T_i}]_{\geq k_i + 1} = [r_{T_i}]_{\geq k_i + 1} ; M' \mid_{\sigma'} \models_{\Pi} [r_{T_i}]_{\geq k_i + 1} \text{ if and only if } \min_{\pi' \in \Pi} \int_{\xi} r_{T_i} [k_i + 1] (\xi') d\Pr_{M', \pi'}^{\sigma, \pi'} \geq -p_i. \) Since for each path \( \xi' \in \mathcal{I} \), \( r_{T_i} [k_i + 1] (\xi) = -1 \) if there exists \( l < k_i + 1 \) such that \( b(\xi')(l) \in T_i \), \( r_{T_i} [k_i + 1] (\xi') = 0 \) otherwise, by the way \( \Pi' \) and \( \sigma' \) are defined, it follows that \( \min_{\pi' \in \Pi} \int_{\xi} r_{T_i} [k_i + 1] (\xi') d\Pr_{M', \pi'}^{\sigma, \pi'} = -\max_{\pi \in \Pi} \Pr_{M, \pi}^{\sigma, \pi} \{ \xi \in \mathcal{I} \mid \exists l \leq k: \xi[l] \in T_i \}. \) Since, by hypothesis, we have that \( \phi \) is satisfiable in \( M \), then it follows that \( \max_{\pi' \in \Pi} \Pr_{M', \pi'}^{\sigma, \pi} \{ \xi \in \mathcal{I} \mid \exists l \leq k: \xi[l] \in T_i \} \leq p_i \), thus \( \min_{\pi' \in \Pi} \int_{\xi} r_{T_i} [k_i + 1] (\xi') d\Pr_{M', \pi'}^{\sigma, \pi'} \geq -p_i \) holds as well, hence \( M' \mid_{\sigma'} \models_{\Pi} [r_{T_i}]_{\geq k_i + 1} = [r_{T_i}]_{\geq k_i + 1} \text{ is satisfied, as required.} \)

This completes the analysis of the case \( \phi' \mid_{h} = \{ [r_{h}]_{\geq k_h} \} \) for each \( i \in \{ 1, \ldots, n \} \).

Let \( h \in \{ n + 1, \ldots, m \} \) and consider \( \phi' \mid_{h} = \{ [r_{h}]_{\geq k_h} \} \); there are two cases depending on the original bound \( \sim_h \).

If \( \sim_h = \geq \), then \( \{ [r_{h}]_{\geq k_h} \} \) holds if and only if \( \min_{\pi' \in \Pi} \int_{\xi} r_{h} [k_h] (\xi) d\Pr_{M', \pi'}^{\sigma, \pi'} \geq r_h \) holds. Since for each path \( \xi' \in \mathcal{I} \), \( \hat{r}_{h} [k_h] \hat{\xi}(\xi') = \hat{r}_{h} [k_h] (b(\xi')) \), by the way \( \Pi' \) and \( \sigma' \) are defined, it follows that \( \min_{\pi' \in \Pi} \int_{\xi} r_{h} [k_h] (\xi) d\Pr_{M', \pi'}^{\sigma, \pi'} = -\max_{\pi \in \Pi} \int_{\xi} r_{h} [k_h] (\xi) d\Pr_{M, \pi}^{\sigma, \pi}. \) Since by hypothesis \( \phi \) is satisfiable in \( M \), then it follows that \( \max_{\pi \in \Pi} \int_{\xi} r_{h} [k_h] (\xi) d\Pr_{M, \pi}^{\sigma, \pi} \leq r_h \), thus \( \min_{\pi' \in \Pi} \int_{\xi} r_{h} [k_h] (\xi) d\Pr_{M', \pi'}^{\sigma, \pi'} \geq -r_h \) holds as well, hence \( M' \mid_{\sigma'} \models_{\Pi} [r_{h}]_{\geq k_h} = [r_{h}]_{\geq k_h} \) is satisfied, as required.

Consider now the second case: If \( \sim_h = \leq \), then \( \{ [r_{h}]_{\geq k_h} \} \) holds if and only if \( \min_{\pi' \in \Pi} \int_{\xi} r_{h} [k_h] (\xi) d\Pr_{M', \pi'}^{\sigma, \pi'} \geq -r_h \) holds. Since for each path \( \xi' \in \mathcal{I} \), \( \hat{r}_{h} [k_h] \hat{\xi}(\xi') = -\hat{r}_{h} [k_h] (b(\xi')) \), by the way \( \Pi' \) and \( \sigma' \) are defined, it follows that \( \min_{\pi' \in \Pi} \int_{\xi} r_{h} [k_h] (\xi) d\Pr_{M', \pi'}^{\sigma, \pi'} = -\max_{\pi \in \Pi} \int_{\xi} r_{h} [k_h] (\xi) d\Pr_{M, \pi}^{\sigma, \pi}. \) Since by hypothesis \( \phi \) is satisfiable in \( M \), then it follows that \( \max_{\pi \in \Pi} \int_{\xi} r_{h} [k_h] (\xi) d\Pr_{M, \pi}^{\sigma, \pi} \leq r_h \), thus \( \min_{\pi' \in \Pi} \int_{\xi} r_{h} [k_h] (\xi) d\Pr_{M', \pi'}^{\sigma, \pi'} \geq -r_h \) holds as well, hence \( M' \mid_{\sigma'} \models_{\Pi} [r_{h}]_{\geq k_h} = [r_{h}]_{\geq k_h} \) is satisfied, as required.

This completes the analysis of the case \( \phi' \mid_{h} = \{ [r_{h}]_{\geq k_h} \} \) for each \( h \in \{ n + 1, \ldots, m \} \); since \( M' \mid_{\sigma'} \models_{\Pi} \phi' \) for each \( j \in \{ 1, \ldots, m \} \), it follows that \( \phi \) is satisfiable in \( M' \), as required to prove that "if \( \phi \) is satisfiable in \( M \), then \( \phi' \) is satisfiable in \( M' \)."

Suppose now the other implication, namely "if \( \phi' \) is satisfiable in \( M' \), then \( \phi \) is satisfiable in \( M \)" and assume that \( \phi' \) is satisfiable in \( M' \): By definition, it follows that there exists a strategy \( \sigma' \) of \( M' \) such that \( M' \mid_{\sigma'} \models_{\Pi} \phi' \); that is, \( M' \mid_{\sigma'} \models_{\Pi} [r_{T_i}]_{\geq k_i + 1} \) for each \( i \in \{ 1, \ldots, n \} \) and \( M' \mid_{\sigma'} \models_{\Pi} [r_{h}]_{\geq k_h} \) for each \( h \in \{ n + 1, \ldots, m \} \). Let \( \sigma \) be the strategy of \( M \) such that, for each finite path \( \xi \in \mathcal{I} \) and action \( a \in \mathcal{A} \), \( \xi' \mid_{\sigma} (a) = \sigma' (\hat{\xi'} (a), v) \) otherwise, where \( (a, v) = \nu_{\sigma'} (last (\hat{\xi'}), a) \). Intuitively, \( \sigma \) chooses the next action \( a \) exactly as \( \sigma' \) chooses \( a \), since \( v \) is uniquely determined by \( \hat{\xi} \). We claim that \( \sigma \) is such that \( M \mid_{\sigma} \models_{\Pi} \phi \).

Let \( i \in \{ 1, \ldots, n \} \) and consider \( \phi_i = [T_i]_{\geq k_i} \); there are two cases depending on the bound \( \sim_i \).

If \( \sim_i = \geq \), then \( M \mid_{\sigma} \models_{\Pi} [T_i]_{\geq k_i} \) if and only if \( \min_{\pi \in \Pi} \Pr_{M, \pi}^{\sigma, \pi} \{ \xi \in \mathcal{I} \mid \exists l \leq k: \xi[l] \in T_i \} \geq p_i. \) Since for each path \( \xi \in \mathcal{I} \), \( r_{T_i} [k_i + 1] (\hat{\xi}) = 1 \) if there exists \( l < k_i + 1 \) such that \( \xi[l] \in T_i \),
Consider now the second case: If \( \sim_i = \leq \), then \( M|_i = \bigcap \{ T_i \}_{i \leq k} \) if and only if \( \max_{\pi \in \Pi} \Pr^{\sigma, \pi}_{M} = \{ \xi \in IPaths \mid \exists l \leq k: \xi[l] \in T_l \} \leq p_i \). Since for each path \( \xi \in Paths, r_{T_i}[k_i + 1](\xi) = -1 \) if there exists \( l < k_i + 1 \) such that \( \xi[l] \in T_l \). \( r_{T_i}[k_i + 1](\xi) = 0 \) otherwise, by the way \( I \) and \( \sigma \) are defined, it follows that \( \max_{\pi \in \Pi} \Pr^{\sigma, \pi}_{M} = \{ \xi \in IPaths \mid \exists l \leq k: \xi[l] \in T_l \} \leq -\min_{\pi \in \Pi} \int_{\xi} r_{T_i}[k_i + 1](\xi) \Pr^{\sigma, \pi}_{M}. \) Since by hypothesis \( \phi' \) is satisfiable in \( M' \), it follows that \( \min_{\pi \in \Pi} \int_{\xi} r_{T_i}[k_i + 1](\xi) \Pr^{\sigma, \pi}_{M} \geq p_i \), thus \( \Pr^{\sigma, \pi}_{M} = \{ \xi \in IPaths \mid \exists l \leq k: \xi[l] \in T_l \} \leq p_i \) holds as well, hence \( M|_i = \bigcap \{ T_i \}_{i \leq k} = \bigcap \{ T_i \}_{i \leq k} \) is satisfied, as required.

This completes the analysis of the case \( \phi_i = [T_i]_{i \leq k} \) for each \( i \in \{1, \ldots, n\} \).

Let \( h \in \{n + 1, \ldots, m\} \) and consider \( \phi_h = [r_h]_{h \leq k} \). There are two cases depending on the original bound \( \sim_h \).

If \( \sim_h = \geq \), then \( M|_i = \bigcap \{ r_h \}_{h \sim_r} \) if and only if \( \min_{\pi \in \Pi} \int_{\xi} r_h[k_h](\xi) \Pr^{\sigma, \pi}_{M} \geq r_h \). Since for each path \( \xi \in Paths, \hat{r}_h[k](\xi) = r_h[k](\xi), \) by the way \( I' \), \( r_h \), and \( \sigma \) are defined, it follows that \( \min_{\pi \in \Pi} \int_{\xi} r_h[k_h](\xi) \Pr^{\sigma, \pi}_{M} = \min_{\pi \in \Pi} \int_{\xi} \hat{r}_h[k_h](\xi) \Pr^{\sigma, \pi'}_{M}. \) Since by hypothesis \( \phi' \) is satisfiable in \( M' \), then \( \min_{\pi \in \Pi} \int_{\xi} \hat{r}_h[k_h](\xi) \Pr^{\sigma, \pi'}_{M} \geq r_h \), thus \( \min_{\pi \in \Pi} \int_{\xi} r_h[k_h](\xi) \Pr^{\sigma, \pi}_{M} \geq r_h \) holds as well, hence \( M|_i = \bigcap \{ r_h \}_{h \sim_r} = [r_h]_{h \sim_r} \) is satisfied, as required.

Consider now the second case: If \( \sim_h = \leq \), then \( M|_i = \bigcap \{ r_h \}_{h \sim_r} \) if and only if \( \max_{\pi \in \Pi} \int_{\xi} r_h[k_h](\xi) \Pr^{\sigma, \pi}_{M} \leq r_h \). Since for each path \( \xi \in Paths, \tilde{r}_h[k](\xi) = r_h[k](\xi), \) by the definition of the components \( I' \), \( \tilde{r}_h \), and \( \sigma \), it is the case that \( \max_{\pi \in \Pi} \int_{\xi} r_h[k_h](\xi) \Pr^{\sigma, \pi}_{M} = -\min_{\pi \in \Pi} \int_{\xi} \tilde{r}_h[k_h](\xi) \Pr^{\sigma, \pi'}_{M}. \) Since by hypothesis \( \phi' \) is satisfiable in \( M' \), then \( \min_{\pi \in \Pi} \int_{\xi} \tilde{r}_h[k_h](\xi) \Pr^{\sigma, \pi'}_{M} \geq r_h \), thus \( \max_{\pi \in \Pi} \int_{\xi} r_h[k_h](\xi) \Pr^{\sigma, \pi}_{M} \leq r_h \) holds as well, hence \( M|_i = \bigcap \{ r_h \}_{h \sim_r} = [r_h]_{h \sim_r} \) is satisfied, as required.

This completes the analysis of the case \( \phi_h = [r_h]_{h \sim_r} \) for each \( h \in \{n + 1, \ldots, m\} \); since \( M|_i = \bigcap \{ \phi_j \}_{j \in \{1, \ldots, m\}} \) for each \( j \in \{1, \ldots, m\} \), it follows that \( \phi \) is satisfiable in \( M \) as required to prove that "if \( \phi' \) is satisfiable in \( M' \), then \( \phi \) is satisfiable in \( M \)." Having proved both implications, the statement of the proposition "\( \phi \) is satisfiable in \( M \) if and only if \( \phi' \) is satisfiable in \( M' \)" holds, as required.

\[ \square \]

**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 \( r \), \( \sup(\ExpTot^{\sigma, \omega}_{M}[r] \bigcap M|_i \bigcap \{ T_i \}_{i \leq k}) = \infty \). From Lemma 14, it follows that if state-action pair \( (s, a) \) occurs infinitely often, \( s \) and \( a \) are contained in a SEC \( E_M \). Therefore, to satisfy the assumed condition, there must exist some strategy \( \sigma \) such that \( M|_i \bigcap \{ T_i \}_{i \leq k} \) and a SEC is reachable, in which \( \sigma \) picks action \( a \) at reachable state \( s \) with positive probability, and \( r(s, a) > 0 \).

**Direction \( \Leftarrow \).** Assume that there is a strategy \( \sigma \) such that \( M|_i \bigcap \{ T_i \}_{i \leq k} \) and a SEC is reachable, and \( r(\xi[n], \xi(n)) > 0 \), where \( \xi \) is a finite path of length \( n + 1 \)
under \( \sigma \) with \( \xi[n] \in S' \) and \( \xi(n) \in A'(\xi[n]) \) 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 \( \{T_n^{\xi_k} \triangleq \emptyset, \ldots, [T_n]^{\pi_k} \} \) are satisfied and (ii) \( \lim_{k \to \infty} \text{ExpTot}^{\sigma_k, k}[r] = \infty \).

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

- for the paths that do not have the prefix \( \xi \), \( \sigma_k \) emulates \( \sigma \).
- when the path \( \xi \) is performed, \( \sigma_k \) forces the system to stay in \( E_M \) containing \( (s, a) \). After \( k \) occurrences of \( (s, a) \), the next time \( s \) is visited, the strategy \( \sigma_k \) emulates \( \sigma \) again as if the performed path segment after \( \xi[n] \) was never executed.

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

(ii) To show that \( \lim_{k \to \infty} \text{ExpTot}^{\sigma_k, k}[r] = \infty \), recall that the probability of reaching \( (s, a) \) under \( \sigma_k \) 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, \( \text{ExpTot}^{\sigma_k, k}[r] \geq p_1 p_2 \frac{r}{s}(s, a) \), and hence, \( \lim_{k \to \infty} \text{ExpTot}^{\sigma_k, k}[r] = \infty \).

\[ \square \]

REFERENCES


Received January 2018; revised September 2018; accepted January 2019