Novel Approximation Bounds Based on Bisimulations for Probabilistic Model Checking of Markov Chains

Jack Wheatley
Balliol College
University of Oxford

A thesis submitted for the degree of
Master of Science in Computer Science

Supervised by
Prof. Alessandro Abate

Trinity Term, 2017

Abstract

In the field of verification, model checking is a technique used to decide the correctness of systems via a mathematical model, usually a transition system. We will be studying systems that exhibits stochastic behaviour, and hence our focus is on probabilistic model checking. In practice, it can be computationally intractable to use probabilistic model checking algorithms when the state space of the model is very large. In this case, we can use approximate probabilistic bisimulations (APBs) to further abstract this model and reduce the state space, while preserving a robust class of properties of the original model. The class of properties that are preserved depends on the approximation error of the abstraction, and it is this error we wish to address.

In this project we propose a new abstraction for reducing the state space of large, labelled Markov chains, that leverages the semantics of uncertain Markov processes – in particular interval Markov decision processes – and the existing notion of APBs. In doing so we produce a model that in general has reduced one-step abstraction error when compared to a like-sized APB, and we give bounds on the propagation of this error in time that also outperform similarly derived error propagation bounds for the classical abstraction. We also outline existing methods for performing model checking via this new approach, and show that the computational complexity of this process is comparable to that for model checking via APBs.

Acknowledgements

First and foremost, thank you to my supervisor, Professor Alessandro Abate. I could not have produced this project without your enthusiasm for the work I have done, your knowledge of the topic, and your close and thoughtful guidance.

Thank you to (the soon to be Doctor) Yuriy Zacchia Lun for your equally zealous co-supervision, and your invaluable, careful comments and proofreading.

Thank you to my family, especially my parents, Justine and Richard. I am so grateful for your endless support in whatever I want to do, for allowing me to study here in Oxford for another year, and for putting up with me whenever I visit home.

Finally, a huge thank you to all of my friends here at Oxford who have supported me throughout this year and the writing of this project, and who have made this the most wonderful of the five years I have spent at Oxford. This project is dedicated to Sage and Alex.

Contents

List of Figures

Introduction

Background and motivation

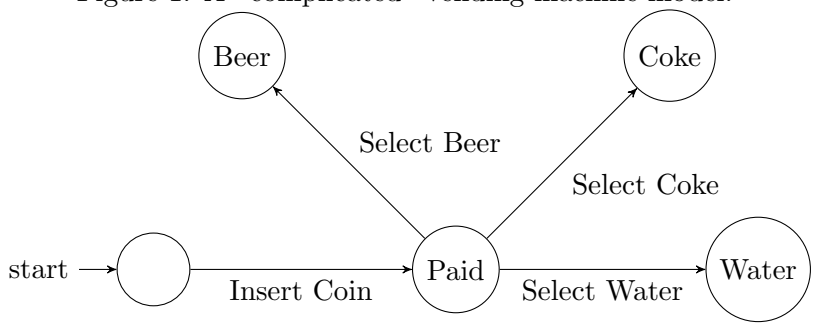
In this thesis we develop a new method for probabilistic model checking of large state space probabilistic systems. Model checking is a technique used in formal verification, a field which concerns itself with establishing the correctness of physical or digital systems. The broad term "systems" includes software, protocols, algorithms, as well as hardware, chemical reaction networks, and other physical networks. Model checking is a way of applying mathematical rigor to the process of system verification. It involves abstracting such a system to a state-based model that "decrib[es] the possible system behaviour in a mathematically precise and unambiguous manner" ([1], p7), and then using formal mathematical methods to verify or provide a counterexample to the correctness of properties of the model expressed in some formal language. The sorts of property that are often of interest include ensuring the system never reaches a deadlock, the system having a low probability of reaching certain "bad" states, and the probability of reaching a desired set of states within a time-bound. The models are usually labelled transition systems, and the exact nature of the model depends on the kind of behaviour exhibited by the system we are modelling.

For general transition systems, the formalisation of the property to check can be done so using a temporal logic, such as linear temporal logic or computational tree logic. If the system in question exhibits memoryless stochastic behaviour (memoryless meaning that the probabilities depend only on the state we are in, not the history of the states we have already visited), then a natural formal description of the system would be a Markov process, and properties may be expressed in probabilistic computational tree logic (PCTL), an extension of computational tree logic [2]. Furthermore, depending on the system in question, we can consider modelling it as a discrete-time, or continuous-time Markov process. In the latter case we would express properties of the model in a continuous-time logic, such as continuous stochastic logic [3]. See [1] for broad introduction to model checking and the logics mentioned above.

Model checking procedures for PCTL properties of Markov processes, which are the focus of this work, have been implemented in the tool PRISM [4]. Model checking is a brute force approach to verification that explores the entire state space, and the algorithms for

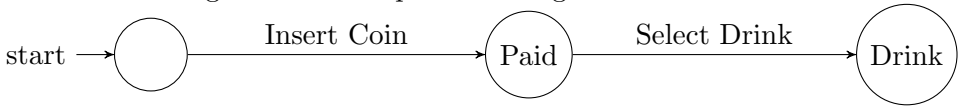
checking a given specification (property) in general run in time at least polynomial in the number of states of the model. This means that in reality, the model checking procedures can be computationally intractable for models with very large state spaces. In this case, we must further abstract away from the system we are modelling, and provide a description of it that is coarser – has a smaller state space. One received method of performing this abstraction is via bisimulation. Roughly, two states are bisimular if they exhibit the same behaviour. As a trivial analogy, consider modelling the procedure of ordering a drink from a vending machine. We might include multiple states for each of the possible drinks one could choose from, as in Figure 1.

Figure 1: A "complicated" vending machine model.



However, depending on our specification, we may not care about which particular drink we receive, and hence we could reduce all the individual drink states to a single state as in Figure 2. Bisimularity is an equivalence relation between states, which is often induced by the labelling of the states in the model, and by the incoming or outgoing transitions. In this analogy, the bisimular states are all those labelled with drinks.

Figure 2: A "simpler" vending machine model.



Finding bisimular states is an important tool for model checking, as properties expressed in certain logics are preserved between the original model and its abstraction under the bisimulation relation. This holds true in the case of stochastic models, where there is a notion of probabilistic bisimularity between states – which is a relation on states that have the same labelling and same outgoing probabilistic behaviour.

However, there are two cases in which using probabilistic bisimulation to reduce the state space of a stochastic model runs into trouble. The first is where the outgoing probabilities from like-labelled states are very similar in behaviour, but not exactly equal. The second stems from the fact that for many probabilistic models, the transition probabilities

are determined experimentally and hence are only determinable up to a degree of accuracy. In both these cases, an "exact" probabilistic bisimulation would not be appropriate, and we would not have the same guarantees on the preservation of all properties that is the motivation for using bisimulations. Thus there is a notion of "approximate" probabilistic bisimulation, where we determine that the states in the approximate bisimularity relation are within some degree of error away from each other (the precise definitions of this will be discussed in due course). This error (or precision) is typically denoted by ε, and it is known that under approximate probabilistic bisimulation, a class – dependent on ε – of "robust" properties are preserved by the abstracted model [5]. Approximate probabilistic bisimulations (APBs) are well studied, and algorithms exist for finding the maximal APB – maximal with respect to the number of approximately bisimular states – of a given error [6]. If we want to find an APB with smaller error, we must expect the abstracted model to have a larger state space. If the error can be reduced, the class of properties that are preserved by the abstraction is larger, which is desirable. However, there is a trade-off between finding an APB with reduced error, and increased computation time due to the larger state space. It is this trade-off we want to address.

Contribution

Often in practice, one is a priori given a lumping (or partition) of the state space of a discrete-time Markov chain (DTMC), usually dictated by the labelling of the states. One existing approach to applying APBs to the problem of model checking DTMCs with a given lumping is to choose a single representative of each class of similarly labelled states that best represents the lump – i.e. that has the least error when considering the states in the lump as being approximately bisimular. In this thesis, we develop a new approach to creating abstractions of DTMCs with given lumpings. Our approach relies on creating a set of virtual representatives (not necessarily states of the concrete model) of each lump, and leveraging the semantics of uncertain Markov chains and Markov decision processes. In doing so, we produce an model that in general has reduced error compared to any possible abstraction obtained via the received method described above, and yet is comparable in terms of the computation time required to perform probabilistic model checking over this model.

Outline

This work is organised as follows: in the first chapter we give a background of the models and properties of interest, and provide a survey of various existing approaches to interpreting and presenting uncertainty in Markov processes. We consider two semantical interpretations of uncertainty in Markov chains, one in which the environment resolves the uncertainty before the process begins, and another where the environment resolves the

uncertainty online at each time-step. We go onto study the relationship between interval, polytopic, and parametric approaches to modelling uncertainty. Then the same considerations are made for Markov decision processes, and similar relationships are drawn.

In the second chapter, we explore how we can find abstracted models with reduced error by examining different choices of virtual stochastic vectors. First, we consider only those vectors that lie within the convex hull of the vectors corresponding to the states we are abstracting, and then develop our new abstraction – the virtual interval Markov decision process (vIMDP) – which considers the transition set of all vectors with best possible error. We then consider a case study that shows how to convert a DTMC with a given lumping into its corresponding vIMDP.

In the following chapters we give an analysis of our new method, first exploring how the computation complexity of model checking using our new approach compares to that of the old one. Initially, the bounds on complexity of our method appear to be exponentially worse than the received method, but work in [7] shows that polynomial time algorithms for model checking interval Markov decision processes exist, potentially making our method computationally feasible. We then provide a geometric view of what our model looks like in $\mathbb{R}^3$ under different possible circumstances, in order to give the reader a more intuitive understanding of this new method. Finally, we compare various bounds on how the error in our new model propagates for certain properties compared to the old model. We develop two bounds on the PCTL safety property error, one that grows linearly in the associated error of our construction and one that grows negative-exponentially. We return to our initial case study to demonstrate that these propagated error bounds for our new model are improvements on the corresponding bounds for the old approach. Lastly, we exploit the Bellman equations for PCTL bounded until to provide a conservative linearly-growing bound on the error for bounded until properties, and again use the case study to show our new model continues to provide improvements over the existing approach. For all of these case studies, the models were implemented and checked in PRISM. We include the specifications of the models and properties checked in Appendix A.

Chapter 1

Background

In this chapter we first give an introduction to the models used for the verification of stochastic processes, namely Markov chains and Markov decision processes. Markov chains are memoryless, probabilistic models, and Markov decision processes extend Markov chains by allowing for non-determinism to be introduced, which is assumed to be resolved by either the environment, or by some agent or adversary. All models considered in this work are discrete-time. We further present the aspects of model checking of Markov processes that are most of interest to this thesis; probabilistic computational tree logic, and the notion of approximate probabilistic bisimulation.

Secondly, this chapter gives a broad overview of existing approaches to modelling uncertainty in Markov processes. Our eventual aim in this work is to leverage the semantics and models of uncertain Markov chains for model checking large state space Markov chains. Hence we present a survey and comparison of such techniques here, so that we may choose that which best suits our needs in later chapters.

1.1 Definitions

1.1.1 Probability space

Definition 1. Let Ω be an arbitrary non-empty set. A *σ-algebra* on Ω is a collection Σ of subsets of Ω that is closed under complementation and countable union, i.e.:

- If $A \in \Sigma$, then $\Omega \setminus A \in \Sigma$,
- For any countable indexing set I, if for all $i \in I$, $A_i \in \Sigma$, then $\bigcup_{i \in I} A_i \in \Sigma$,
- $\emptyset \in \Sigma$.

Definition 2. A *probability space* is a triple $(\Omega, \Sigma, \mathbf{Pr})$ where:

- Ω is a non-empty set called the sample space,
- Σ is a σ-algebra on Ω called the collection of events,
- $\mathbf{Pr} : \Sigma \to [0, 1]$ is the probability measure, that satisfies the following:
  - $\mathbf{Pr}(\Omega) = 1$,
  - for any countable indexing set I, $\mathbf{Pr}(\bigcup_{i \in I} A_i) = \sum_{i \in I} \mathbf{Pr}(A_i)$, if the A_i are disjoint.

1.1.2 Markov processes

Definition 3. A *discrete-Time Markov chain* (DTMC) is a tuple $\mathcal{D} = (Q, q_0, P, L)$, where:

- Q is a non-empty, finite set of states,
- $q_0 \in Q$ is the initial state,
- $P : Q \times Q \to [0, 1]$ is a stochastic matrix of transition probabilities,
- $L : Q \to 2^{AP}$ is a labelling function, where AP is a fixed set of atomic propositions, or labels.

For a finite set of states Q, we will write $\Delta(Q)$ for the set of distributions over Q; the set of functions $\mu : Q \to [0, 1]$ such that $\sum_{s \in Q} \mu(s) = 1$. Equivalently, we can consider $\Delta(Q)$ as a set of stochastic vectors in $\mathbb{R}^{|Q|}$. We let P_s denote the row of the matrix P corresponding to the transitions from state $s \in Q$, which will be a member of $\Delta(Q)$.

Definition 4. An *infinite path* in a DTMC $\mathcal{D} = (Q, q_0, P, L)$ is an infinite sequence of states $\omega = s_0, s_1, \ldots$ where for all $i \in \mathbb{N}$, $s_i \in Q$ and $P(s_i, s_{i+1}) > 0$. We write $\omega(i)$ for the $(i + 1)$-th state on the path ω, and write $Paths^{\mathcal{D}}(s)$ for the set of all infinite paths in $\mathcal{D}$ such that $\omega(0) = s$.

A *finite path* in a DTMC is defined likewise, except that $\omega = s_0, s_1, \ldots, s_n$ for some $n \in \mathbb{N}$. We denote the set of all finite paths begin at state s by $Paths^{\mathcal{D}}_{fin}(s)$. Where it is clear which model is being considered we will omit the superscript $\mathcal{D}$ from $Paths^{\mathcal{D}}(s)$ and $Paths^{\mathcal{D}}_{fin}(s)$.

Definition 5. For a finite path ω in a DTMC $\mathcal{D} = (Q, q_0, P, L)$, we define the *cylinder set*, $Cyl(\omega) := \{\omega' \in Paths(s) \mid \omega \text{ is a prefix of } \omega'\}$.

For a given DTMC $\mathcal{D} = (Q, q_0, P, L)$, the transition probabilities given by the matrix P can be extended to determine the probability of specific finite paths unfolding from a given state s. For any finite path $\omega = s, s_1, \ldots, s_n$ in $\mathcal{D}$ starting at s we have the function:

$$P_s(\omega) := \begin{cases} 1 & \text{if the length of } \omega \text{ is one} \\ P(s, s_1) \cdot \ldots \cdot P(s_{n-1}, s_n) & \text{otherwise} \end{cases}$$

We hence define a probability space over all paths beginning from a state s of the DTMC as follows:

- The sample space is $\Omega = Paths(s)$,
- The collection of events is $\Sigma_{Paths(s)}$, which is defined to be the least σ-algebra on $Paths(s)$ containing $Cyl(\omega)$ for all finite paths ω starting at s,
- The probability measure $\mathbf{Pr}_s : \Sigma_{Paths(s)} \to [0, 1]$ is defined by $\mathbf{Pr}_s(Cyl(\omega)) = P_s(\omega)$ for any finite path ω, which uniquely extends to a probability measure on the whole event space.

For further details on probability spaces over paths in Markov chains, see [8].

Definition 6. A *Markov decision process* (MDP) is a tuple $\mathcal{M} = (Q, q_0, \delta, L)$, where:

- Q, q_0 and L are defined as for DTMCs,
- $\delta : Q \to 2^{\Delta(Q)}$ is a transition function such that for all $s \in Q$, the set $\delta(s)$ is finite and non-empty.

Remark 1. We will refer to each $\delta(s)$ as the *set of actions at* s. Where needed, we will also label these actions, though we leave this addition labelling function out of the formal definition for brevity of notation.

An infinite path in an MDP $\mathcal{M} = (Q, q_0, \delta, L)$ is a sequence of states and actions, $\omega = s_0, \mu_0, s_1, \mu_1, \ldots$ where $\forall i \in \mathbb{N}$, $s_i \in Q, \mu_i \in \delta(s_i)$ and $\mu_i(s_{i+1}) > 0$. We again write $\omega(i)$ for the $(i+1)$-th state on the path ω, and write $Paths(s)$ for the set of all paths in $\mathcal{M}$ such that $\omega(0) = s$. We further write $Paths_{fin}(s)$ for the set of all finite paths beginning at s.

An MDP allows for probabilistic processes with non-deterministic choices at each state. The non-determinism is characterised by the set of actions at each state of the model. The non-determinism is then resolved by either an environmental or agential choice at each state.

Definition 7. Given an MDP $\mathcal{M} = (Q, q_0, \delta, L)$, an *adversary* (or scheduler, strategy, or policy) is a function $\sigma : Paths_{fin}(s) \to \Delta(Q)$ for a state $s \in Q$ such that for any finite path $\omega = s, \mu_0, s_1, \mu_1, \ldots, s_n$, $\sigma(\omega) \in \delta(s_n)$.

We write $Paths^{\sigma}(s)$ for the infinite paths from s where non-determinism has been resolved by an adversary σ, i.e. paths $s, \mu_0, s_1, \mu_1, \ldots$ where $\forall n \in \mathbb{N}, \sigma(s, \mu_0, \ldots, s_n) = \mu_n$. An adversary for an MDP induces an infinite state DTMC:

Definition 8. Given an MDP $\mathcal{M} = (Q, q_0, \delta, L)$, and an adversary $\sigma : Paths_{fin}(s) \to \Delta(Q)$, the *infinite DTMC induced by* σ is $\mathcal{M}^{\sigma} = (Paths^{\sigma}_{fin}(s), s, P^{\sigma}, L^{\sigma})$, where:

- $P^\sigma(\omega, \omega') = \begin{cases} \mu(s') & \text{if } \omega' = \omega, \mu, s' \text{ and } \sigma(\omega) = \mu \\ 0 & \text{otherwise} \end{cases}$
- $L^\sigma(s_0, \mu_0, \ldots, s_n) = L(s_n)$.

In general this induced DTMC is not strictly a DTMC as we have defined it above, as it can have an infinite state space. However, for a certain class of adversaries the induced DTMC can be represented by a finite-state DTMC:

Definition 9. Given an MDP $\mathcal{M} = (Q, q_0, \delta, L)$, a state $s \in Q$, and an adversary $\sigma : Paths_{fin}(s) \to \Delta(Q)$, we say that σ is memoryless if and only if for any two sequences $\omega_1 = s, \mu_0, s_1, \mu_1, \ldots, s_n$, and $\omega_2 = s, \mu_0', t_1, \mu_1', \ldots, t_m$ in $Paths_{fin}(s)$ with $s_n = t_m$, then $\sigma(\omega_1) = \sigma(\omega_2)$.

In other words, an adversary is memoryless if and only if the choice of action it makes at a state is always the same, independent of which states have already been visited. In this case, for an MDP $\mathcal{M} = (Q, q_0, \delta, L)$, the adversary can equivalently be written as a function:

$$\sigma : Q \to \bigcup_{s \in Q} \delta(s),$$

where for each $s \in Q$, $\sigma(s) \in \delta(s)$, i.e. as a mapping of states to an action at that state. In this case, the induced DTMC *is* a finite-state DTMC $\mathcal{M}^\sigma = (Q, q_0, \tilde{P}, L)$, where $\tilde{P}$ is a transition probability matrix in which the row corresponding to $s \in Q$ is $\sigma(s)$.

1.1.3 Probabilistic computational tree logic

We are interested in the verification of probabilistic properties of the above processes. For DTMCs and MDPs, we usually express these properties in Probabilistic Computational Tree Logic, PCTL. The syntax and semantics of PCTL for DTMCs is defined here:

Definition 10. *PCTL syntax*:

- $\phi ::= true \mid a \mid \phi \wedge \phi \mid \neg\phi \mid \mathbb{P}_{\sim p}[\psi]$ (state formulae)
- $\psi ::= \mathbf{X}\phi \mid \phi \mathbf{U}^{\leq k} \phi \mid \phi \mathbf{U} \phi$ (path formulae)

where a is an atomic proposition, $p \in [0, 1]$ is a probability bound, $\sim \in \{<, >, \leq, \geq\}$, and $k \in \mathbb{N}$. A PCTL formula is always defined to be a *state* formula.

Definition 11. *PCTL semantics for DTMCs*:
Given a DTMC $\mathcal{D} = (Q, q_0, P, L)$, and a state $s \in Q$:

- $s \vDash true$ always,
- $s \vDash a \Leftrightarrow a \in L(s)$,

- $s \vDash \phi_1 \wedge \phi_2 \Leftrightarrow s \vDash \phi_1$ and $s \vDash \phi_2$,
- $s \vDash \neg\phi \Leftrightarrow s \nvDash \phi$,
- $s \vDash \mathbb{P}_{\sim p}[\psi] \Leftrightarrow \mathbf{Prob}(s, \psi) := \mathbf{Pr}_s\{\omega \in \mathrm{Path}(s) \mid \omega \vDash \psi\} \sim p$.

For a path ω in $\mathcal{D}$:

- $\omega \vDash \mathbf{X}\phi \Leftrightarrow \omega(1) \vDash \phi$,
- $\omega \vDash \phi_1 \mathbf{U}^{\leq k} \phi_2 \Leftrightarrow \exists 0 \leq i \leq k$ such that $\omega(i) \vDash \phi_2$ and $\forall 0 \leq j < i\ \omega(j) \vDash \phi_1$,
- $\omega \vDash \phi_1 \mathbf{U} \phi_2 \Leftrightarrow \exists k \geq 0$ such that $\omega(k) \vDash \phi_2$ and $\forall 0 \leq j < k\ \omega(j) \vDash \phi_1$.

We further will write $\mathcal{D} \vDash \phi$ if $q_0 \vDash \phi$.

We will also use the following common short-hands for path formulae:

- $\mathbf{F}\phi \equiv true\mathbf{U}\phi$ and $\mathbf{F}^{\leq k}\phi \equiv true\mathbf{U}^{\leq k}\phi$,
- $\mathbf{G}\phi \equiv \neg\mathbf{F}\neg\phi$ and $\mathbf{G}^{\leq k}\phi \equiv \neg\mathbf{F}^{\leq k}\neg\phi$.

Strictly, $\mathbf{G}\phi$ (and $\mathbf{G}^{\leq k}\phi$) is not derivable from the syntax of PCTL path formulae, but the state formulae we are interested in – i.e. of the form $\mathbb{P}_{\sim p}[\mathbf{G}\phi]$ – are directly derivable in PCTL as:

$$\begin{aligned} s \vDash \mathbb{P}_{\leq p}[\mathbf{G}\phi] &\Leftrightarrow \mathbf{Prob}(s, \mathbf{G}\phi) \leq p \\ &\Leftrightarrow 1 - \mathbf{Prob}(s, \mathbf{G}\phi) \geq 1 - p \\ &\Leftrightarrow \mathbf{Prob}(s, \neg\mathbf{G}\phi) \geq 1 - p \\ &\Leftrightarrow \mathbf{Prob}(s, \mathbf{F}\neg\phi) \geq 1 - p \\ &\Leftrightarrow s \vDash \mathbb{P}_{\geq 1-p}[true\mathbf{U}\neg\phi], \end{aligned}$$

and $\mathbb{P}_{\geq 1-p}[true\mathbf{U}\neg\phi]$ is a valid PCTL state formula.

Definition 12. Given a DTMC $\mathcal{D} = (Q, q_0, P, L)$ and a PCTL state formula ϕ, we write:

$$Sat(\phi) = \{s \in Q \mid s \vDash \phi\}$$

For an MDP $\mathcal{D} = (Q, q_0, \delta, L)$, a state $s \in Q$, a path formula ψ and an adversary σ, let us write $\mathbf{Prob}^\sigma(s, \psi) := \mathbf{Pr}_s\{\omega \in Paths^\sigma(s) \mid \omega \vDash \psi\}$. We further write $p_{\min}(s, \psi) := \inf_\sigma \mathbf{Prob}^\sigma(s, \psi)$ and $p_{\max}(s, \psi) := \sup_\sigma \mathbf{Prob}^\sigma(s, \psi)$.

Definition 13. *PCTL semantics for MDPs*:
Given an MDP $\mathcal{M} = (Q, q_0, \delta, L)$ and a state s of $\mathcal{M}$:

- For all non-probabilistic state formulae, and path formulae the semantics are identical to those for DTMCs,

- if $\sim \in \{\geq, >\}$, then $s \vDash \mathbb{P}_{\sim p}[\psi] \Leftrightarrow p_{\min}(s, \psi) \sim p$,
- if $\sim \in \{\leq, <\}$, then $s \vDash \mathbb{P}_{\sim p}[\psi] \Leftrightarrow p_{\max}(s, \psi) \sim p$

As with DMTCs, we will write $\mathcal{M} \vDash \phi$ if $q_0 \vDash \phi$. Note further that $\mathcal{M} \vDash \phi$ if and only if for all adversaries σ at q_0, the DTMC $\mathcal{M}^\sigma \vDash \phi$.

Theorem 1. Given an MDP $\mathcal{M} = (Q, q_0, \delta, L)$, any state $s \in Q$ and a PCTL path formula ψ, there exist memoryless adversaries $\sigma_{\min}$ and $\sigma_{\max}$ such that $\mathbf{Prob}^{\sigma_{\min}}(s, \psi) = p_{\min}(s, \psi)$ and $\mathbf{Prob}^{\sigma_{\max}}(s, \psi) = p_{\max}(s, \psi)$.

Proof. See [1]. □

For both DTMCs and MDPs, in addition to verifying whether a PCTL state formula holds at a state in the model, we can also query the probability of the model satisfying a path formula, i.e. all of the values $\mathbf{Prob}(s, \psi)$, $p_{\min}(s, \psi)$, and $p_{\max}(s, \psi)$ are calculable via various existing model checking algorithms. For DTMCs, we write this query as:

$$\mathbb{P}_{=?}[\psi]$$

for a given path formula, ψ, and for MDPs we have:

$$\mathbb{P}_{max=?}[\psi] \text{ and } \mathbb{P}_{max=?}[\psi].$$

1.1.4 Approximate probabilistic bisimulation

For a DTMC (Q, q_0, P, L), we will write $P^k(s, s')$ for the probability that a state s' is reached in k steps from s, and for a set $A \subseteq Q$, we let $P^k(s, A) := \sum_{s' \in A} P^k(s, s')$. If the set $A \subseteq Q$ can be written as $Sat(\phi)$ for any PCTL state formula ϕ, then $P^k(s, A)$ can equivalently be written as $\mathbf{Prob}(s, \mathbf{X}^k \phi)$, where:

$$\mathbf{X}^k \phi = \underbrace{\mathbf{X} \ldots \mathbf{X}}_{k \text{ times}} \phi$$

Definition 14. Given a relation $\Gamma \subseteq Q \times Q$, we say that $A \subseteq Q$ is Γ-*closed* if:

$$\Gamma(A) = \{s \in Q \,|\, \exists s' \in A \text{ such that } (s, s') \in \Gamma\} \subseteq A$$

Definition 15. Given an DTMC $\mathcal{D} = (Q, q_0, P, L)$, a *(exact) probabilistic bisimulation* is an equivalence relation $\Gamma \subseteq Q \times Q$ such that for any $(s, s') \in \Gamma$, $L(s) = L(s')$, and for any Γ-closed $A \subseteq Q$, $P(s, A) = P(s', A)$.

Definition 16. Given an DTMC $\mathcal{D} = (Q, q_0, P, L)$, an *approximate probabilistic bisimulation* (APB) with precision/error ε is a reflexive, symmetric relation $\Gamma_\varepsilon \subseteq Q \times Q$ such that for any $(s, s') \in \Gamma_\varepsilon$, $L(s) = L(s')$, and for any Γ_ε-closed $A \subseteq Q$, $|P(s, A) - P(s', A)| \leq \varepsilon$.

Definition 17. Given a DTMC $\mathcal{D} = (Q, q_0, P, L)$ and an APB Γ_ε, we define the *probabilistic realisation distance* induced by Γ_ε at time $k \geq 1$, and for $(s, s') \in \Gamma_\varepsilon$:

$$d^k_{\Gamma_\varepsilon}(s, s') := \max_{A:\, \Gamma_\varepsilon - closed} |P^k(s, \Gamma_\varepsilon(A)) - P^k(s', \Gamma_\varepsilon(A))|$$

and we further define $d^\infty_{\Gamma_\varepsilon}(s, s') := \lim_{k\to\infty} d^k_{\Gamma_\varepsilon}(s, s')$, and $d^k_{\Gamma_\varepsilon}(\mathcal{D}) := \max_{(s,s')\in\Gamma_\varepsilon} d^k_{\Gamma_\varepsilon}(s, s')$.

While approximate bisimulations will not ensure the preservation of all PCTL formulae, in [5] it is proven that the existence of an ABP with error ε implies the preservation of a certain subclass of PCTL formulae; namely those that are ε-robust. We define this notion here:

Definition 18. Given an DTMC $\mathcal{D} = (Q, q_0, P, L)$, a PCTL formula ϕ, and an APB Γ_ε with error ε, the *ε-strengthened PCTL formula* $S_\varepsilon(\phi)$ and the *ε-weakened PCTL formula* $R_\varepsilon(\phi)$ are defined inductively as follows:

1. $S_\varepsilon(true) = true$,
   $R_\varepsilon(true) = true$.
2. $S_\varepsilon(a) = a$,
   $R_\varepsilon(a) = a$.
3. $S_\varepsilon(\neg\phi) = \neg S_\varepsilon(\phi)$,
   $R_\varepsilon(\neg\phi) = R_\varepsilon(\phi)$.
4. $S_\varepsilon(\phi \wedge \psi) = S_\varepsilon(\phi) \wedge S_\varepsilon(\psi)$,
   $R_\varepsilon(\phi \wedge \psi) = R_\varepsilon(\phi) \wedge R_\varepsilon(\psi)$.
5. $S_\varepsilon(\mathbb{P}_{\sim p}[X\phi]) = \mathbb{P}_{\sim p'}[X S_\varepsilon(\phi)]$, where $p' = \begin{cases} p - \varepsilon & \text{if } \sim \in \{<, \leq\} \\ p + \varepsilon & \text{if } \sim \in \{>, \geq\} \end{cases}$,
   $R_\varepsilon(\mathbb{P}_{\sim p}[X\phi]) = \mathbb{P}_{\sim p'}[X R_\varepsilon(\phi)]$, where $p' = \begin{cases} p + \varepsilon & \text{if } \sim \in \{<, \leq\} \\ p - \varepsilon & \text{if } \sim \in \{>, \geq\} \end{cases}$.
6. For any $k \in \mathbb{N} \cup \{\infty\}$:
   $S_\varepsilon(\mathbb{P}_{\sim p}[\phi U^{\leq k} \psi]) = \mathbb{P}_{\sim p'}[S_\varepsilon(\phi) U^{\leq k} S_\varepsilon(\psi)]$, where $p' = \begin{cases} p - d^k_{\Gamma_\varepsilon}(\mathcal{D}) & \text{if } \sim \in \{<, \leq\} \\ p + d^k_{\Gamma_\varepsilon}(\mathcal{D}) & \text{if } \sim \in \{>, \geq\} \end{cases}$,
   $R_\varepsilon(\mathbb{P}_{\sim p}[\phi U^{\leq k} \psi]) = \mathbb{P}_{\sim p'}[R_\varepsilon(\phi) U^{\leq k} R_\varepsilon(\psi)]$, where $p' = \begin{cases} p + d^k_{\Gamma_\varepsilon}(\mathcal{D}) & \text{if } \sim \in \{<, \leq\} \\ p - d^k_{\Gamma_\varepsilon}(\mathcal{D}) & \text{if } \sim \in \{>, \geq\} \end{cases}$.

Definition 19. Given a DTMC $\mathcal{D} = (Q, q_0, P, L)$, PCTL formula ϕ, and an APB Γ_ε with error ε, we say ϕ is *ε-robust (with respect to $\mathcal{D}$)* if for any $s \in Q$ and any subformula ψ of ϕ, either $s \in Sat(S_\varepsilon(\psi))$, or $s \notin Sat(R_\varepsilon(\psi))$.

We have the following theorem, that shows that ε-robust formulae are indeed preserved by models obtained from the original by an approximate bisimulation of precision ε:

Theorem 2. Given a DTMC $\mathcal{D} = (Q, q_0, P, L)$, an APB Γ_ε, and a PCTL formula ϕ that is ε-robust with respect to M, then for all $s, s' \in Q$ such that $(s, s') \in \Gamma_\varepsilon$ the following holds:

$$s \in Sat(\phi) \Leftrightarrow s' \in Sat(\phi).$$

1.2 Uncertainty in Markov chains

1.2.1 Markov set-chains

DTMCs as formulated above are time-homogeneous; the transition matrix is constant across all time-steps. The study of Markov set-chains is motivated by the recognition that time-homogeneity is not suitable for all applications. Thus we want to consider time-non-homogeneous Markov chains, where the transition matrix may change in time. Markov set-chains are one such formulation of this non-homogeneity that allow for fluctuating transition matrices but maintain much of the theory of homogeneous Markov chains. From [9] we obtain the following definition:

Definition 20. Let $\mathcal{S}$ be a compact set of $n \times n$ stochastic matrices, and let Q be a finite set of states. A *Markov set-chain* is the sequence $(\mathcal{S}^k)_{k>0}$, where:

$$\mathcal{S}^k := \{P_1 P_2 \dots P_k \,|\, \forall i = 1, \dots k \, P_i \in \mathcal{S}\}.$$

Examples of compact sets of stochastic matrices are transition sets, and convex hulls of finitely many stochastic matrices, both of which will be explored in due course. The set $\mathcal{S}^k$ for a Markov set-chain describes all the possible behaviours of the transition system after k time-steps.

1.2.2 Interval Markov chains

We obtain the following definitions from [10]:

Definition 21. An *interval Markov Chain* (IMC) is a tuple $\mathcal{I} = (Q, q_0, P^l, P^u, L)$ where:

- Q, q_0 and L are defined as for DTMCs,
- $P^l, P^u : Q \times Q \to [0, 1]$ are matrices such that $P^l \leq P^u$ (where $\leq$ is element-wise) where $P^l(s, s')$ (resp. $P^u(s, s')$) gives the lower (resp. upper) bound of the transition probability from state s to s'.

We might also want to consider an IMC in the following way: for $\mathcal{I} = (Q, q_0, P^l, P^u, L)$, with matrices $P^l = (l_{ij}), P^u = (u_{ij}) \in \mathbb{R}^{|Q|\times|Q|}$, we can represent $\mathcal{I}$ by the $|Q| \times |Q|$ matrix of intervals $([l_{ij}, u_{ij}])$.

There are two semantic interpretations of IMCs that we will consider here, uncertain Markov chains (UMCs), and interval Markov decision processes (IMDPs). In the following we will consider the IMC $\mathcal{I} = (Q, q_0, P^l, P^u, L)$.

In the UMC semantics, we take $\mathcal{I}$ to represent an infinite set of DTMCs; the set of all DTMCs (Q, q_0, P, L) such that:

$$\forall s, s' \in Q,\ P^l(s, s') \leq P(s, s') \leq P^u(s, s').$$

Definition 22. For any $m, n \in \mathbb{N}^+$, given $P, T \in \mathbb{R}^{m\times n}$, non-negative matrices such that $P \leq T$ (where $\leq$ is element-wise), a *transition set* is defined to be:

$$[\Pi] = [P, T] := \{W \in \mathbb{R}^{m\times n} \,|\, W \text{ is stochastic and } P \leq W \leq T\}.$$

Transition sets expressed by different intervals may contain the same stochastic matrices. When we say that two transition sets are equal, we will mean that they contain exactly the same elements, and write this as $[P_1, T_1] = [P_2, T_2]$.

We can hence write the IMC $\mathcal{I}$ as $(Q, q_0, [\Pi], L)$, where $[\Pi]$ is the transition set $[P^l, P^u]$. Under the UMC semantics, we assume an element $P \in [\Pi]$ is chosen non-deterministically by the external environment at the start of the process, and the behaviour of the transition system from then on is described by the DTMC (Q, q_0, P, L). Formally we define a UMC of this sort as follows:

Definition 23. Given an IMC $\mathcal{I} = (Q, q_0, [\Pi], L)$, we define the *UMC corresponding to* $\mathcal{I}$ to be the set of all DTMCs (Q, q_0, P, L) such that $P \in [\Pi]$.

In the IMDP semantics, we assume that the behaviour at each time-step is determined by a different non-deterministic choice of matrix $P \in [\Pi]$ by the environment. This bears close resemblance to a Markov decision process, and hence the nomenclature "interval *MDP*" for the semantics induced by $\mathcal{I}$. We give a formal definition for this notion:

Definition 24. Given an IMC $\mathcal{I} = (Q, q_0, [\Pi], L)$, define the *IMDP corresponding to* $\mathcal{I}$ to be the tuple $\tilde{\mathcal{I}} = (Q, q_0, \delta, L)$ where:

- Q, q_0 and L are defined as for DTMCs,
- $\delta : Q \to 2^{\Delta(Q)}$ such that for any $s \in Q$, $\delta(s) = \{\mu \in \Delta(Q) \,|\, \forall s' \in Q, P^l(s, s') \leq \mu(s') \leq P^u(s, s')\}$.

In this setting, it is usually assume that the transition probability matrix is chosen non-determin-istically by the environment, not by an agent. If we assume that all we know is the transition set $[\Pi]$ from which matrices can be chosen, then this setting is the same

as a Markov set-chain, as all we can determine about the behaviour of the process at time k is that the probability matrix falls somewhere in $[\Pi]^k$, as defined for MSCs. We make the distinction between IMDPs and classical MDPs as in the IMDP case, for any $s \in Q$, $\delta(s)$ may contain infinitely many distributions.

PCTL semantics for IMCs

Given an IMC $\mathcal{I} = (Q, q_0, [\Pi], L)$, a PCTL formula ϕ, when considering the UMC semantics for IMCs, we will write $\mathcal{I} \vDash \phi$ if and only if for all $P \in [\Pi]$, the DTMC $\mathcal{D} = (Q, q_0, P, L)$ is such that $\mathcal{D} \vDash \phi$.

Furthermore, when considering the IMDP semantics for IMCs, we interpret $\tilde{\mathcal{I}} \vDash \phi$ in the same way as we interpret the relation $\vDash$ for classical MDPs. In [11], it is shown that PCTL model checking for IMDPs can be reduced to PCTL model checking for MDPs:

Theorem 3. Given an IMDP $\tilde{\mathcal{I}} = (Q, q_0, \delta, L)$, there exists an MDP $\mathcal{M}$ such that for any PCTL formula ϕ, $\tilde{\mathcal{I}} \vDash \phi$ if and only if $\mathcal{M} \vDash \phi$.

1.2.3 Polytopic Markov chains

For IMCs, we take our transition probability matrix to lie within some interval defined by two matrices. Here we give a generalised version of this notion, where the transition probability matrices all lie within the convex polytope defined by two or more matrices, as seen in [12].

Definition 25. For any $m, n \in \mathbb{N}^+$, given a set of matrices $\mathcal{P} = \{P_1, \ldots, P_k\}$ all in $\mathbb{R}^{m \times n}$, the *convex hull of* $\mathcal{P}$ or *the convex polytope with vertices in* $\mathcal{P}$ is $conv(\mathcal{P}) :=$

$$\{P \in \mathbb{R}^{m \times n} \mid P = \lambda_1 P_1 + \cdots + \lambda_k P_k, \text{ where } \forall 1 \leq i \leq k, \lambda_i \geq 0, \text{ and } \sum \lambda_i = 1\}.$$

Definition 26. A *polytopic Markov Chain* (polyMC) is a tuple $\mathcal{PO} = (Q, q_0, \mathcal{P}^K, L)$ where:

- Q, q_0 and L are defined as for DTMCs,
- $\mathcal{P}^K = \{P_i : Q \times Q \to [0, 1] \mid i = 1, \ldots, K\}$, a finite set of stochastic matrices such that $|\mathcal{P}^K| > 1$ and whose elements are the vertices of a convex polytope.

As for IMCs, there are two possible semantic interpretations of polyMCs. The first corresponds to the UMC semantics, and the second to the IMDP semantics in natural ways, except here matrices are chosen non-deterministically from a convex polytope, rather than an interval.

Definition 27. Given a polyMC $\mathcal{PO} = (Q, q_0, \mathcal{P}^K, L)$, we define the *UMC corresponding to* $\mathcal{PO}$ to be the set of all DTMCs (Q, q_0, P, L) such that $P \in conv(\mathcal{P}^K)$.

Definition 28. Given a polyMC $\mathcal{PO} = (Q, q_0, \mathcal{P}^K, L)$, define the *polyMDP corresponding to* $\mathcal{PO}$ to be the tuple $\widetilde{\mathcal{PO}} = (Q, q_0, \gamma, L)$ where:

- Q, q_0 and L are defined as for DTMCs,
- $\gamma : Q \to 2^{\Delta(Q)}$ such that for any $s \in Q$, $\gamma(s) = \{\mu \in \Delta(Q) \mid \mu = P_s \text{ for some } P \in conv(\mathcal{P}^K)\}$.

Furthermore, the PCTL semantics for IMCs naturally carry over and are identically defined for polyMCs. The following propositions prove ther is a strict inclusions of IMCs among polyMCs:

Proposition 1. Every IMC can be represented as a polyMC.

Proof. We sketch a proof based on results from [9], which we will give in full detail in a later chapter. In this work it is shown that any transition set $[P, T]$ is a convex polytope and its vertices can be determined. As all the elements of a transition set are stochastic, then so too are the vertices. Hence, given an IMC $\mathcal{I} = (Q, q_0, [\Pi], L)$, we can determine the vertex set of the polytope $[\Pi]$, which we denote $\mathcal{P}^{\mathcal{I}}$, and obtain a polyMC $\mathcal{PO}^{\mathcal{I}} = (Q, q_0, \mathcal{P}^{\mathcal{I}}, L)$, that is equivalent to the IMC $\mathcal{I}$. □

Proposition 2. polyMCs cannot in general be represented by an IMC.

Proof. Take for example the convex polytope defined by $\mathcal{P} = \{\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{smallmatrix}\right)\}$. An interval $[P, T]$ containing both these matrices would have to be such that:

$$P \leq \begin{pmatrix} 0.9 & 0 \\ 0 & 0.9 \end{pmatrix} \text{ and } \begin{pmatrix} 1 & 0.1 \\ 0.1 & 1 \end{pmatrix} \leq T,$$

in which case $\left(\begin{smallmatrix} 1 & 0 \\ 0.1 & 0.9 \end{smallmatrix}\right) \in [P, T]$, which is not a member of $conv(\mathcal{P})$. Therefore there is no interval that can represent the polytope $conv(\mathcal{P})$. □

1.2.4 Parametric Markov chains

For polyMCs, and hence IMCs, any possible transition matrix can be represented as a linear sum of the K vertices of the corresponding convex polytope. Specifically, for polyMC $\mathcal{PO} = (Q, q_0, \mathcal{P}^K, L)$, where $\mathcal{P}^K = \{P_1, \ldots, P_K\}$, the set of possible transition matrices for this polyMC to choose from can be represented by the following matrix of parametric equations, with parameters $\lambda_1, \ldots, \lambda_K$:

$$\sum_{m=1}^{K} \lambda_m P_m = (\sum_{m=1}^{K} \lambda_m P_{ij}^m) \text{ where } \forall m = 1, \ldots, K \ \lambda_m \geq 0 \text{ and } \sum_{m=1}^{K} \lambda_m = 1 \tag{1.1}$$

In this section, we introduce a further generalisation of this notion, that of a parametric Markov chain [13]. We will confine our discussion to parametric models parametrised

by rational functions, as this is what typically appears in practice. We define a rational function over a vector of variables $\bar{v} = (v_1, \ldots, v_n)$ to be any function $f(\bar{v}) = \frac{p(\bar{v})}{q(\bar{v})}$, where p, q are polynomials in $\bar{v}$.

Definition 29. A *parametric Markov chain* (pMC) is a tuple $\mathcal{PA} = (Q, q_0, P, V, L)$ where:

- Q, q_0, L are all defined as for DTMCs,
- $V = \{v_1, \ldots, v_k\}$ is a finite set of parameters (variables),
- $P : Q \times Q \to \mathcal{F}_V$ is a transition matrix whose entries are in $\mathcal{F}_V$, the rational functions in the variables among V.

Given a pMC $\mathcal{PA} = (Q, q_0, P, V, L)$, we want to consider the set of evaluations $u : V \to \mathbb{R}$. Given an evaluation u, we write P^u for the real-valued matrix obtained by evaluating each element of P under u in the natural way, i.e. $u(P_{ij}(v_1, \ldots, v_k)) = P_{ij}(u(v_1), \ldots, u(v_k))$. We say an evaluation u is well-defined for the pMC $\mathcal{PA}$ if the resulting matrix P^u is stochastic. A pMC is thus represented by its set of well-defined evaluations.

As for IMCs and polyMC, the two different semantics can be considered for pMCs, and thus we have the following definitions:

Definition 30. Given a pMC $\mathcal{PA} = (Q, q_0, P, V, L)$, we define the *UMC corresponding to* $\mathcal{PA}$ to be the set of all DTMCs (Q, q_0, P^u, L) such that u is a well-defined evaluation for $\mathcal{PA}$.

Definition 31. Given a pMC $\mathcal{PA} = (Q, q_0, P, V, L)$, define the *pMDP corresponding to* $\mathcal{PA}$ to be the tuple $\tilde{\mathcal{PA}} = (Q, q_0, \gamma, L)$ where:

- Q, q_0 and L are defined as for DTMCs
- $\gamma : Q \to 2^{\Delta(Q)}$ such that for any $q \in Q$,
  $\gamma(s) = \{\mu \in \Delta(Q) \mid \mu = P^u_s$ for some well-defined evaluation $u\}$.

As for polyMCs, the PCTL semantics for IMCs carry over and are identically defined for the different semantic interpretations of pMCs.

We wish to show that there is a natural way of representing any polyMC as a pMC. Defined as in 1.1 above, a polyMC doesn't quite fit the definition of a pMC as there are addition bounds on the space of possible parameters. These constraints are necessary for the parametrised matrix in 1.1 to represent only those matrices within the bounds of the polytope. There are two very reasonable relaxations to the above definition of a pMC that would allow for the above realisation of a polyMC to be directly translated into a pMC:

(a) relaxing the type of function allowed within the matrix P to include piecewise rational functions,

(b) allowing for additions constraints on the parameter space, as in 1.1. It is natural that we would want to put bounds on the space of parameters we wish to consider when looking for well-defined evaluations.

A very simple example would be a parametrised matrix $\left(\begin{smallmatrix} 1 & 0 \\ \lambda & 1-\lambda \end{smallmatrix}\right)$, where we know that λ falls within some interval $[a, b] \subset [0, 1]$. We wouldn't want to have to consider all possible well-defined evaluations in this context. With this in mind, we allow also that any number of constraints of the form $a_1v_1 + \cdots + a_nv_n \leq b$ or $a_1v_1 + \cdots + a_nv_n = b$ be included as part of the model.

In either case we have the following proposition:

Proposition 3. Every polyMC can be represented as a pMC.

If we allow for bounds to be put on the parameter space, then clearly the characterisation of a polyMC given in (1) expresses a polyMC as a pMC. If we allow for piecewise rational functions in the parametrised transition matrix we then have the following proof of the above proposition:

Proof. Given a polyMC $\mathcal{PO} = (Q, q_0, \mathcal{P}^K, L)$ where $\mathcal{P}^K = \{P_1, \ldots, P_K\}$, and parameters $\Lambda = \{\lambda_1, \ldots, \lambda_K\}$, for all $1 \leq i, j \leq |Q|$ we define the piecewise rational function $\phi_{ij}(\lambda_1, \ldots, \lambda_K)$ as follows:

$$\phi_{ij}(\lambda_1, \ldots, \lambda_K) = \begin{cases} \sum_{m=1}^{K} \lambda_m P_{ij}^m & \text{if } \forall m = 1, \ldots, K \ \lambda_m \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

This ensures that any evaluation u on Λ is never well-defined if it evaluates any parameter as negative. Furthermore, if we consider an evaluation u that has no negative values, then for any i:

$$\sum_{j=1}^{|Q|} \phi_{ij}(\lambda_1, \ldots, \lambda_K) = \sum_{j=1}^{|Q|} \sum_{m=1}^{K} \lambda_m P_{ij}^m = \sum_{m=1}^{K} \lambda_m \sum_{j=1}^{|Q|} P_{ij}^m = \sum_{m=1}^{K} \lambda_m.$$

An evaluation is well-defined only if this sum is equal to 1, so by necessity we also have the other clause; namely $\sum_{m=1}^{K} \lambda_m = 1$. Hence, it follows that a polyMC can be represented by the pMC $(Q, q_0, (\phi_{ij}), \Lambda, L)$. □

It is then also of interest whether or not the contraposition of the above proposition can be proven, i.e. can any pMC be expressed as a polyMC. The answer to this is no:

Proposition 4. pMCs cannot in general be expressed by polyMCs.

Proof. Consider the pMC defined by the parametrised matrix $\left(\begin{smallmatrix} v^2 & \frac{11}{9} - v \\ 1 & 0 \end{smallmatrix}\right)$. The well-defined evaluations, u, for this pMC are those that satisfy the equation $u(v)^2 + \frac{11}{9} - u(v) = 1$, i.e.

the solutions to this quadratic equation, which are $u(v) = \frac{1}{3}, \frac{2}{3}$ (these are easily checkable to be well-defined!). Thus the set of matrices defined by this pMC is:

$$\{P = \begin{pmatrix} \frac{4}{9} & \frac{5}{9} \\ 1 & 0 \end{pmatrix}, Q = \begin{pmatrix} \frac{1}{9} & \frac{8}{9} \\ 1 & 0 \end{pmatrix}\}.$$

Any convex polytope of matrices that contains both these matrices must also contain any linear combination $\lambda_1 P + \lambda_2 Q$ where $\lambda_1, \lambda_2 \geq 0$ and $\lambda_1 + \lambda_2 = 1$, and hence contains more than two elements. Thus there is no polyMC that represents this pMC. □

We have hence now established that the following "inclusions" hold and are strict:

$$\text{IMC} \subset \text{polyMC} \subset \text{pMC}.$$

1.3 Uncertainty in Markov decision processes

In the previous section we saw that one semantic interpretation of the uncertain transition probabilities in a Markov chain led to a structure closely resembling that of classical MDP, namely what we referred to as IMDPs, polyMDPs, and pMDPs. Here, we look directly at uncertainty in MDPs, rather than constructing MDP-like structures from uncertainty in DTMCs.

1.3.1 Bounded-parameter Markov decision processes

In the MDP-like semantics for the various uncertain Markov chains above, we understood the non-determinism at each state to be completely environmental; the environment chooses a distribution that then determines probabilistically what the next state will be. For what is known in the literature as a bounded-parameter MDP (introduced in [14]) at each state an adversary chooses one of a finite number of possible actions, and then the uncertainty is introduced; the environment non-deterministically chooses a distribution from a set dependent on the choice of the adversary.

Definition 32. A *bounded-parameter Markov decision process* (BMDP) is a tuple $\mathcal{B} = (Q, q_0, \gamma, L)$ such that:

- Q, q_0, and L are defined as for MDPs,
- $\gamma : Q \to 2^{\mathcal{R}}$, where $\mathcal{R} = \mathbb{R}^{|Q|} \times \mathbb{R}^{|Q|}$, is a transition function, such that for any $s \in Q$, $\gamma(s)$ is a non-empty finite set of non-empty transition sets $[p^l, p^u]$.

At each state, when an action is chosen by an adversary, the distribution of that action is only known to be within some interval, for which the upper and lower bounds are given. When these upper and lower bounds are equal for all actions available at all states, the model is a classical MDP.

As with all the models looked at in the previous section, there are two possible semantic interpretations of BMDPs. In the first, the distribution of each action is chosen by the environment at the beginning (a UMC-like interpretation), and the second is as described in the literature for BMDPs; the environment chooses a distribution each time an action is chosen by an adversary at a state (a IMDP-like interpretation). We give a formal definition of the former, which we will call an uncertain MDP in keeping with the naming in the previous section, but we will assume the second interpretation as the standard for BMDPs, and continue refer to the model in this interpretation as such.

Definition 33. Given a BMDP $\mathcal{B} = (Q, q_0, \gamma, L)$, we define the *UMDP corresponding to* $\mathcal{B}$ to be the set of all MDPs (Q, q_0, δ, L) such that for each $s \in Q$ there exists a bijection $\phi_s : \delta(s) \to \gamma(s)$ such that for all pairs induced by this correspondence, $(\mu, \phi_s(\mu)) = (\mu, (p^l, p^u))$, we have that $\mu \in [p^l, p^u]$.

Under the second semantic interpretation, a BMDP is identical in structure to a controlled model introduced in [15], where it is given the name "controlled Markov set-chain". A path in a BMDP is hence a sequence of states and distribution intervals $\omega = s_0, [p_0^l, p_0^u], s_1, [p_1^l, p_1^u], \ldots$ where $\forall i \in \mathbb{N}$, $s_i \in Q$, $(p_i^l, p_i^u) \in \gamma(s_i)$ and the entries corresponding to s_{i+1} in p_i^l and p_i^u are both > 0.

Moreover, in the case where for all $s \in Q$, $|\gamma(s)| = 1$, i.e. when the adversary has no choice of action at any state, then the model is equivalent to the IMDP induced by an IMC. This is clear from the definitions of IMDPs induced by an IMC and BMDPs.

Remark 2. As IMDPs induced by an IMC are a special case of BMDPs, we will abuse the nomenclature somewhat and simply refer to BMDPs as IMDPs, so that the naming is consistent in what follows.

1.3.2 Generalisations of interval Markov decision processes

As in the case of Markov chains in the previous section, we can adapt the above IMDP model to allow for less constrained sets of possible distributions at each state and action. IMDPs relate to IMCs, in that the possible distributions lie within a transition set. It is therefore natural that we also consider uncertain MDPs where the possible distributions the environment can choose from lie within a convex polytope (thus corresponding to polyMCs) and where the distributions are given parametrically (corresponding to pMCs). We would then expect to have a similar hierarchy of inclusions among these models as we had for the various uncertain Markov chain models. We distinguish here between the polyMDP induced by a polyMC and a polyMDP, and likewise between the pMDP induced by a pMC and a pMDP:

Definition 34. A *polytopic Markov decision process* (polyMDP) is a tuple $\mathcal{PO} = (Q, q_0, \gamma, L)$ such that:

- Q, q_0, and L are defined as for MDPs,
- $\gamma : Q \to 2^{2^{\mathcal{R}}}$, where $\mathcal{R} = \mathbb{R}^{|Q|}$, is a transition function, such that for any $s \in Q$, $\gamma(s)$ is a finite set $\{P_q^1, \dots P_q^{K_q}\}$, such that for all $1 \leq i \leq K_q$, P_s^i is a non-empty set of stochastic vectors that are the corners of a convex polytope.

Definition 35. A *parametric Markov decision process* (pMDP) is a tuple $\mathcal{PA} = (Q, q_0, \gamma, V, L)$ such that:

- Q, q_0, and L are defined as for MDPs,
- $V = \{v_1, \dots, v_n\}$ is a finite set of variables,
- $\gamma : Q \to 2^F$, where $F = \mathcal{F}_V^{|Q|}$, is a transition function, such that for any $s \in Q$, $\gamma(s)$ is a non-empty set of vectors of rational functions that have at least 1 well-defined evaluation.

To justify the repetition of names for these models, we can immediately see that an xMDP induced by an xMC is simply an xMDP where $|\gamma(s)| = 1$ for all $s \in Q$ ($x \in$ {I, poly, p}).

1.3.3 Convex uncertainties

In [7], a general model that encompasses both IMDPs and polyMDPs is introduced, named Convex MDPs (CMDPs).

Definition 36. A *convex Markov decision process* (CMDP) is a tuple $\mathcal{C} = (Q, q_0, \gamma, L)$ such that:

- Q, q_0, and L are defined as for MDPs,
- $\beta : Q \to 2^{2^{\mathcal{R}}}$, where $\mathcal{R} = \mathbb{R}^{|Q|}$, is a transition function, such that for any $s \in Q$, $\gamma(s)$ is a finite set $\{C_s^1, \dots C_s^{K_s}\}$, such that for all $1 \leq i \leq K_s$, C_s^i is a non-empty convex set of stochastic vectors.

While not-mentioned in this particular paper, we could of course define a convex Markov chain (CMC), described by the data (Q, q_0, C, L) where C is a convex set of transition probability matrices, and hence obtain the two semantic interpretations: UCMC induced by the CMC, and CMDP induced by the CMC. In which case, again we would also see that a CMDP induced by a CMC is a special case of a CMDP as defined here.

It is important to note the following:

Proposition 5. All IMDPs and polyMDPs can be represented as CMDPs.

This is clear, as the uncertainties in both IMDPs and polyMDPs are described by convex sets of stochastic vectors. The same cannot be said for pMDPs, as we saw that parametrised vectors can have only two distinct well-defined evaluations (see proof of Proposition 4). Further to the inclusion of IMDPs among CMDPs, [7] also considers two other models of uncertainty that fall within the scope of CMDPs. There are referred to as the likelihood model and the ellipsoidal model. These are given just for interest, and so we will only give a brief introduction to these models of uncertainty.

The likelihood model

This model is useful for analysis of systems where the transition probabilities for each choice of action are determined experimentally. At a state $s \in Q$ where the adversary has k choices of action, which we label $a_1, \ldots, a_k$, the transition frequencies associated to each action a_i are recorded in vector H^i. We write $h^i_{s'}$ for the element of H^i corresponding to the frequency of transitions to state s' from s. For each action, we obtain the following convex set of stochastic vectors, called the likelihood region:

$$C^i = \{v \in \Delta(Q) \mid \sum_{s' \in Q} h^i_{s'} \log(v_{s'}) \geq \varepsilon^i_s\},$$

where $\varepsilon^i_s < \sum_{s' \in Q} h^i_{s'} \log(h^i_{s'})$ represents the uncertainty level of the measurements.

The ellipsoidal model

Ellipsoidal models are an approximation of the likelihood model. Here instead:

$$C^i = \{v \in \Delta(Q) \mid \|R^i_s(v - H^i)\|_2, R^i_s \succ 0\},$$

where R^i_s represents an ellipsoidal approximation of the likelihood region above. It is noted that we might allow for different actions in a single CMDP to be expressed with a different uncertainty model in order to allow for different sources of uncertainty within the same system.

1.3.4 MDPs with imprecisely known transition probabilities

Here we consider what has been referred to in the literature (presented in [16]) as MDPs with imprecisely known transition probabilities, or MDPIPs. These is the most general formulation of uncertain MDPs that we will encounter; the transition probabilities for each possible action are just said to be chosen from some set of possible distributions, with no constraint on the properties of that set. Indeed, in [14] it is noted that they introduce IMDPs as a specialisation of the earlier MDPIP.

Definition 37. A *Markov decision process with imprecisely known transition probabilities* (MDPIP) is a tuple $\mathcal{B} = (Q, q_0, \gamma, L)$ such that:

- Q, q_0, and L are defined as for MDPs,
- $\gamma : Q \to 2^{\mathcal{R}}$, where $\mathcal{R} = \mathbb{R}^{|Q|}$, is a transition function, such that for any $s \in Q$, $\gamma(s)$ is a set of stochastic vectors.

We hence have the following:

Proposition 6. Every IMDP, polyMDP, CMDP, and pMDP can be represented as an MDPIP.

Proof. Clear from the definitions. □

A similarly general approach to considering uncertain transition probabilities in MDPs is taken in [17], where they introduce a model they call a "uncertain labelled finite MDP", where the constraint on the transition probabilities for each action are that they must lie within a given "uncertainty set". This is identical in semantics to that of the MDPIP defined above, and even allows for non-cnvex uncertainty sets. In this work, MDPIPs are used to design "robust control policies for an uncertain system subject to temporal logic specifications" ([17], p3372). The adversaries synthesised in this paper maximise the worst case probability of satisfying an LTL property of a system modelled as an uncertain labelled finite MDP.

Chapter 2

A New Approach: Virtual Interval Markov Decision Processes

2.1 The naive approach to constructing abstractions of Markov chains

Take some DMTC $\mathcal{D} = (Q, q_0, P, L)$, and consider a partition of the state space into non-empty subsets $Q_1, \ldots, Q_m$, such that for all $1 \leq i \leq k$, for all $s, s' \in Q_i$, $L(s) = L(s')$. Assume without loss of generality that $q_0 \in Q_1$. This partitioning induces a family of APBs in the following way:

1. Choose an element from each Q_i: $q_1 \in Q_1, \ldots, q_m \in Q_m$.
2. Define a relation on $Q \times Q$: $\Gamma = \bigcup_{Q_i} \{(q_i, s), (s, q_i) \mid s \in Q_i\}$. This is reflexive and symmetric, and all and only the Q_i are Γ-closed. Hence this Γ does define an APB.
3. The error of the APB is then determined by $\max_{(s,s') \in \Gamma} \max_{Q_i} |P(s, Q_i) - P(s', Q_i)|$.

From each of these APBs, Γ, there is a corresponding lumped DTMC $(\mathcal{Q}, Q_1, P_\Gamma, L)$, where $\mathcal{Q} = \{Q_1, \ldots, Q_m\}$. Each Q_i is given the same label as its constituent elements in $\mathcal{D}$. P_Γ is given as follows: $P_\Gamma(Q_i, Q_j) = P(q_i, Q_j)$. This family of APBs contains at most $|Q_1| \times \cdots \times |Q_m|$ elements, each one corresponding to a different combination of choices of one element from each Q_i. A naive approach to reducing the error of an APB for $\mathcal{D}$ is to choose the APB from the family described above that offers the lowest error. It is against this naive approach that we will benchmark our efforts to produce abstractions of DTMCs.

2.2 Developing the new approach

2.2.1 Points in the convex hull

Consider again the DTMC $\mathcal{D} = (Q, q_0, P, L)$ with partition of the state space $Q_1, \ldots, Q_m$. We will focus on the set $Q_1 = s_1, \ldots, s_k$. For each s_i, there is a corresponding vector, $r_i = (P(s_i, Q_1), \ldots, P(s_i, Q_m))$. In the above approach, to reduce the error of the abstraction, we would choose the $s_i \in Q_1$ such that $\varepsilon_i := \max_{j \neq i} \|r_i - r_j\|_\infty$ is minimised. Our first question is: is there some stochastic vector r^* that lies in the convex hull of the vectors $r_1, \ldots, r_k$ such that $\max_{1 \leq i \leq k} \|r^* - r_i\|_\infty < \min_{1 \leq i \leq k} \varepsilon_i$?

Proposition 7. Assume that the $r_1, \ldots, r_k$ are the corners of a convex polytope. Let $r^* = \lambda_1 r_1 + \cdots + \lambda_k r_k$, where for all $1 \leq i \leq k$, $\lambda_i \geq 0$, and $\sum \lambda_i = 1$, so $r^* \in conv(\{r_1, \ldots, r_k\})$. Then for any $1 \leq i \leq k$, the error associated to r^*, $\varepsilon^* \leq \max_{1 \leq i \leq k} (1 - \lambda_i)\varepsilon_i$. Furthermore, if we assume that $\lambda_1 = \cdots = \lambda_k = \frac{1}{k}$, then $\varepsilon^* \leq (\frac{k-1}{k}) \max_{1 \leq i \leq k} \varepsilon_i$.

Proof. For any $1 \leq i \leq k$:

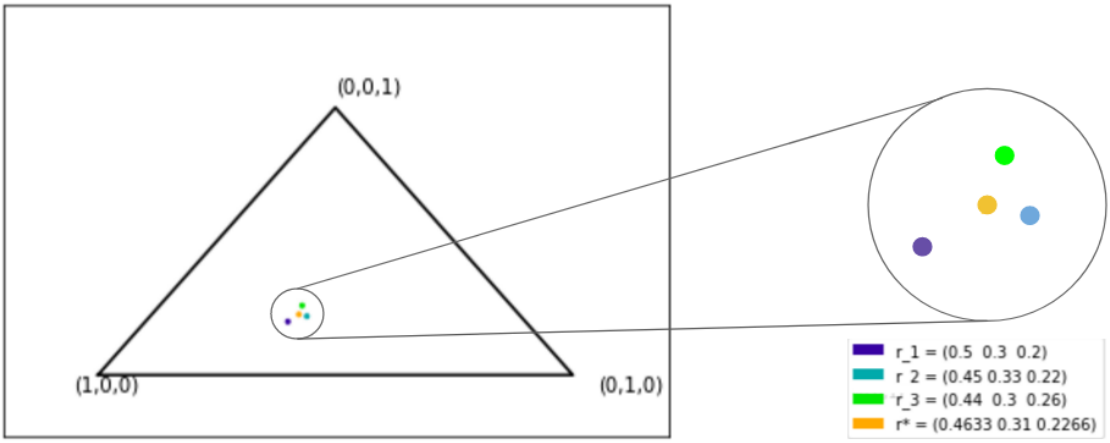
$$\begin{aligned}
\|r^* - r_i\|_\infty &= \|\lambda_1 r_1 + \cdots + \lambda_k r_k - r_i\|_\infty \\
&= \|\lambda_1 (r_1 - r_i) + \cdots + \lambda_k (r_k - r_i)\|_\infty \\
&\leq \sum_{j=1}^{k} \lambda_j \|r_j - r_i\|_\infty \\
&= \sum_{j \neq i} \lambda_j \|r_i - r_j\|_\infty \\
&\leq \sum_{j \neq i} \lambda_j \varepsilon_i \\
&= (1 - \lambda_i)\varepsilon_i
\end{aligned}$$

Therefore, $\varepsilon^* \leq \max_{1 \leq i \leq k} (1 - \lambda_i)\varepsilon_i$. And it follows that if $\lambda_1 = \cdots = \lambda_k = \frac{1}{k}$, then $\varepsilon^* \leq (\frac{k-1}{k}) \max_{1 \leq i \leq k} \varepsilon_i$. □

We will consider for the time being only the case where $\lambda_1 = \cdots = \lambda_k = \frac{1}{k}$, i.e. where r^* is the centre of mass of the $r_1, \ldots, r_k$. Notice that this point only offers an improvement on any of the errors associated to the original points if $\min_i \varepsilon_i > (\frac{k-1}{k}) \max_i \varepsilon_i$.

Example 1. As a trivial example of when this method produces an improved error, suppose $Q_1 = \{q_1, q_2\}$, with associated rows r_1, r_2, respectively. Then if we let $r^* = \frac{1}{2}(r_1 + r_2)$ we have that $\|r^* - r_1\|_\infty = \|r^* - r_2\|_\infty = \|\frac{1}{2}(r_1 - r_2)\|_\infty = \frac{1}{2}\varepsilon_1 = \frac{1}{2}\varepsilon_2$. Taking the centre of mass of the rows reduces the error by a factor of 2.

Figure 2.1: Example 2, $r^* = \frac{1}{3}(r_1 + r_2 + r_3)$.



In the above example, we saw that the above upper bound on ε^* is tight; in some cases $\varepsilon^* = (\frac{k-1}{k}) \max\limits_{1 \leq i \leq k} \varepsilon_i$. Now for an example when $\varepsilon^* < (\frac{k-1}{k}) \max\limits_{1 \leq i \leq k} \varepsilon_i$:

Example 2. Consider rows $r_1 = (0.5, 0.3, 0.2), r_2 = (0.45, 0.33, 0.22), r_3 = (0.44, 0.3, 0.26)$. These rows are the corners of a convex polytope, as they are 3 linearly independent vectors in $\mathbb{R}^3$. For these rows, we have the corresponding errors; $\varepsilon_1 = 0.06, \varepsilon_2 = 0.05, \varepsilon_3 = 0.06$, and it holds that $\min\limits_i \varepsilon_i > (\frac{3-1}{3}) \max\limits_i \varepsilon_i$. Now $r^* = \frac{1}{3}(r_1 + r_2 + r_3) = (0.4633, 0.31, 0.2266)$, and has corresponding error $\varepsilon^* = 0.036\dot{6} < \frac{2}{3} \max\limits_i \varepsilon_i$. See Figure 2.1 for a visual representation.

Now an example when the centre of mass does not offer an improved precision rate:

Example 3. Consider rows $r_1 = (0.5, 0.25, 0.25), r_2 = (0.45, 0.3, 0.25), r_3 = (0.4, 0.3, 0.3)$. These rows are the corners of a convex polytope, as they are 3 linearly independent vectors in $\mathbb{R}^3$. For these rows, we have the corresponding errors; $\varepsilon_1 = 0.1, \varepsilon_2 = 0.05, \varepsilon_3 = 0.1$. Immediately we see that because $0.05 < \frac{3-1}{3} \cdot 0.1 = 0.06\dot{6}$, that the above requirement does not hold of this example. Furthermore, taking $r^* = \frac{1}{3}(r_1 + r_2 + r_3) = (0.45, 0.2833, 0.2666)$ we obtain an error $\varepsilon^* = 0.05$, which is no improvement on ε_2. Furthermore, ε_2 is already the minimum achievable error for any stochastic vector: if for some r^* we have the corresponding error $\varepsilon^* < 0.05$, then $\|r^* - r_1\|_\infty < 0.05$. So $|r_1^* - r_{11}| < 0.05$, and hence $0.35 < r_1^* < 0.45$. In which case $\|r^* - r_3\|_\infty > 0.05$, which means that $\varepsilon^* > 0.05$, a contradiction. So $\varepsilon_2 = 0.05$ is the optimal error in this example.

It is not generally the case, like in Example 3, that if $\min\limits_i \varepsilon_i \leq (\frac{k-1}{k}) \max\limits_i \varepsilon_i$ then $\min\limits_i \varepsilon_i$ is the optimal error achievable:

Example 4. A slight perturbation of the three rows in Example 3 gives a counterexample: consider instead $r_1 = (0.5, 0.25, 0.25), r_2' = (0.45, 0.31, 0.24)$, and $r_3 = (0.4, 0.3, 0.3)$. Again these rows are the corners of a convex polytope for the same reason as in Example 3. We have now the following errors: $\varepsilon_1 = 0.1, \varepsilon_2' = 0.06, \varepsilon_3 = 0.1$, and again because $0.06 < \frac{3-1}{3} \cdot 0.1 = 0.06\dot{6}$, the antecedent of the statement holds. However we know from the previous example, and the fact that $\|r_2 - r_2'\|_\infty = 0.01$, that r_2 would give an better error in this scenario; 0.05.

2.2.2 Characterising the rows with optimal error

Lemma 1. Let $\varepsilon_{max} = \max\limits_{1 \leq i \leq k} \varepsilon_i = \max\limits_{i,j} \|r_i - r_j\|_\infty$. Then, the optimal error achievable by any stochastic vector is $\frac{1}{2}\varepsilon_{max}$.

Proof. Let r be a stochastic vector, and suppose for contradiction that its associated error $\varepsilon < \frac{1}{2}\varepsilon_{max}$. Let i_0, j_0 be such that $\|r_{i_0} - r_{j_0}\|_\infty = \varepsilon_{max}$. Then $\|r - r_{i_0}\|_\infty \leq \varepsilon < \frac{1}{2}\varepsilon_{max}$ and $\|r - r_{j_0}\|_\infty \leq \varepsilon < \frac{1}{2}\varepsilon_{max}$. But $\|r_{i_0} - r_{j_0}\|_\infty = \|(r_{i_0} - r) + (r - r_{j_0})\|_\infty \leq \|r - r_{i_0}\|_\infty + \|r - r_{j_0}\|_\infty < \frac{1}{2}\varepsilon_{max} + \frac{1}{2}\varepsilon_{max} = \varepsilon_{max}$. A contradiction as $\|r_{i_0} - r_{j_0}\|_\infty = \varepsilon_{max}$. So it follows that $\varepsilon \geq \frac{1}{2}\varepsilon_{max}$, as required. □

We can represent the set of stochastic vectors that achieve the optimal error value by a transition set:

Proposition 8. For $r_1, \ldots, r_k$, for $1 \leq j \leq m$, let $u_j := \min\limits_{1 \leq i \leq k} r_{ij}$, and $v_j := \max\limits_{1 \leq i \leq k} r_{ij}$. Now consider the transition set $[u, v] := ([u_1, v_1], \ldots, [u_m, v_m])$. Let $\beta = \frac{1}{2}\varepsilon_{max}$, where ε_{max} is defined as in Lemma 1. Then the family of stochastic vectors r such that $\max\limits_{1 \leq i \leq k} \|r - r_i\|_\infty = \beta$ is exactly the transition set:

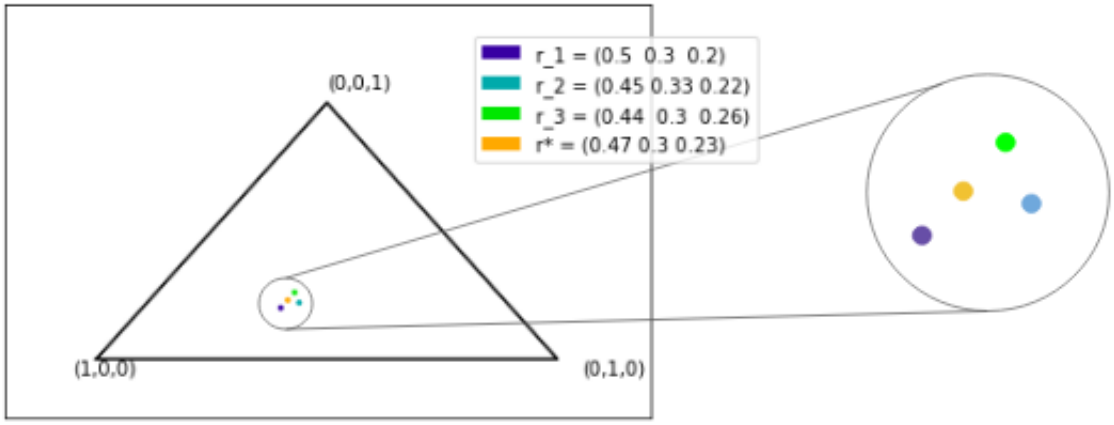
$$[u, v]_{opt} := ([v_1 - \beta, u_1 + \beta], \ldots, [v_m - \beta, u_m + \beta]).$$

Proof. This is clear by construction. □

If $[u, v]_{opt}$ is non-empty, then there is a stochastic vector r_{opt} with associated error $\varepsilon_{opt} = \frac{1}{2}\varepsilon_{max}$. Observe that $conv(\{r_1, \ldots, r_k\}) \subseteq [u, v]$. It is not necessarily the case that $[u, v]_{opt} \subset conv(\{r_1, \ldots, r_k\})$, as if for some $1 \leq i \leq m$, $\beta > v_i - u_i$, then $[u_i, v_i] \subset [v_i - \beta, u_i + \beta]$. Hence we cannot restrict our search to only those vectors that fall within the convex hull of the rows.

Example 5. Let us consider again the rows from Example 2; $r_1 = (0.5, 0.3, 0.2), r_2 = (0.45, 0.33, 0.22)$, and $r_3 = (0.44, 0.3, 0.26)$. We get $[u, v] = ([0.44, 0.5], [0.3, 0.33], [0.2, 0.26])$. We find that $\beta = 0.03$, and hence $[u, v]_{opt} = (0.47, [0.3, 0.33], 0.23) = \{(0.47, 0.3, 0.23)\}$. So letting $r^* = (0.47, 0.3, 0.23)$, we get associated error $\varepsilon^* = \beta = 0.03$, far better compared to the error we achieved in Example 2 of $0.0366\dot{}$ that we obtained by taking the average of

Figure 2.2: Example 5, $[u, v]_{opt}$ here is a single point.



the 3 rows. Further, $r^* = \frac{1}{2}(r_1 + r_3)$, so lies in the $conv(\{r_1, r_2, r_3\})$. See Figure 2.2 for a visual representation.

Here is an example of when we have $[u, v]_{opt} \nsubseteq conv(\{r_1, \ldots, r_k\})$:

Example 6. Let us consider $r_1 = (0.2, 0.3, 0.25, 0.25), r_2 = (0.3, 0.27, 0.22, 0.21)$ and $r_3 = (0.22, 0.27, 0.26, 0.25)$. Then $[u, v] = ([0.2, 0.3], [0.27, 0.3], [0.22, 0.26], [0.21, 0.25])$ and $\beta = 0.05$, so we get $[u, v]_{opt} = (0.25, [0.25, 0.32], [0.21, 0.27], [0.2, 0.26])$.

Now we have $r^* = (0.25, 0.25, 0.25, 0.25) \in [u, v]_{opt}$, but $r^* \notin conv(\{r_1, \ldots, r_k\})$, because $r^*_2 = 0.25 < r_{12}, r_{22}, r_{32}$, and hence there are no non-negative $\lambda_1, \lambda_2, \lambda_3$ such that $\lambda_1 + \lambda_2 + \lambda_3 = 1$ and $\lambda_1 r_{12} + \lambda_2 r_{22} + \lambda_3 r_{32} = r^*_2$.

Lemma 2. It is not always the case that $[u, v]_{opt}$ is non-empty.

Proof. Consider again the rows from Example 2 and Example 5: $r_1 = (0.5, 0.3, 0.2), r_2 = (0.45, 0.33, 0.22)$, and $r_3 = (0.44, 0.3, 0.26)$. If we consider in addition to these the row $r_4 = (0.45, 0.34, 0.21)$, we still have that $\varepsilon_{max} = \|r_1 - r_3\|_\infty = 0.06$, so the optimal achievable error is 0.03. In this case, $[u, v] = ([0.44, 0.5], [0.3, 0.34], [0.2, 0.26])$ and hence $[u, v]_{opt} = (0.47, [0.31, 0.33], 0.23) = \emptyset$. □

A question naturally arises from the result of Lemma 2: can we reduce the error still in the cases where no stochastic vectors with associated error equal to $\beta = \frac{1}{2}\varepsilon_{max}$ exist? In the following we see that by a simple procedure a vector with close to optimal error can be produced in the case where $[p, q]_{opt}$ is empty:

Proposition 9. For any $m \in \mathbb{N}^+$, if for a set of stochastic vectors $\{r_1, \ldots, r_k\}$ all in $\mathbb{R}^m$ the set $[u, v]_{opt}$ corresponding to this set of vectors is empty, then there is a vector r^* such that its corresponding error $\varepsilon^* \leq \max\{\frac{1-\sum_{i=1}^m p_i}{m}, \frac{\sum_{i=1}^m q_i - 1}{m}\}$.

Proof. If $[u,v]_{opt} = ([v_1-\beta, u_1+\beta], \ldots, [v_m-\beta, u_m+\beta])$ is empty, then either $\sum_{i=1}^m (v_i-\beta) > 1$, or $\sum_{i=1}^m (u_i+\beta) < 1$.

Suppose first that $\sum_{i=1}^m (v_i-\beta) > 1$. We know that $r = (v_1-\beta, \ldots, v_m-\beta)$ is non-stochastic and $\max_{1\le i\le k} \|r - r_i\|_\infty \le \beta$. Let

$$\delta = \frac{\sum_{i=1}^m (v_i-\beta) - 1}{m} = \frac{\sum_{i=1}^m v_i - 1}{m} - \beta,$$

then $r^* = (v_1-\beta-\delta, \ldots, v_m-\beta-\delta)$ is stochastic and we have that:

$$\varepsilon^* = \max_{1\le i\le k} \|r^* - r_i\|_\infty \le \beta + \delta = \frac{\sum_{i=1}^m v_i - 1}{m}.$$

Furthermore, if $\sum_{i=1}^m (v_i-\beta) > 1$, then also $\sum_{i=1}^m (u_i+\beta) > 1$. Rearranging these inequalities we get:

$$\frac{1-\sum_{i=1}^m u_i}{m} < \beta < \frac{\sum_{i=1}^m v_i - 1}{m},$$

and hence that:

$$\varepsilon^* \le \max\{\frac{1-\sum_{i=1}^m u_i}{m}, \frac{\sum_{i=1}^m v_i - 1}{m}\}$$

Suppose instead that $\sum_{i=1}^m (u_i+\beta) < 1$. Then let

$$\delta = \frac{1-\sum_{i=1}^m (u_i+\beta)}{m} = \frac{1-\sum_{i=1}^m u_i}{m} - \beta.$$

In this case $r^* = (u_1+\beta+\delta, \ldots, u_m+\beta+\delta)$ is stochastic and

$$\varepsilon^* = \max_{1\le i\le k} \|r - r_i\|_\infty \le \beta + \delta = \frac{1-\sum_{i=1}^m u_i}{m}.$$

Similarly to above, if $\sum_{i=1}^m (u_i+\beta) < 1$, then $\sum_{i=1}^m (v_i-\beta) < 1$, and rearranging gives:

$$\frac{\sum_{i=1}^m v_i - 1}{m} < \beta < \frac{1-\sum_{i=1}^m u_i}{m},$$

and hence:

$$\varepsilon^* \le \max\{\frac{1-\sum_{i=1}^m u_i}{m}, \frac{\sum_{i=1}^m v_i - 1}{m}\}.$$

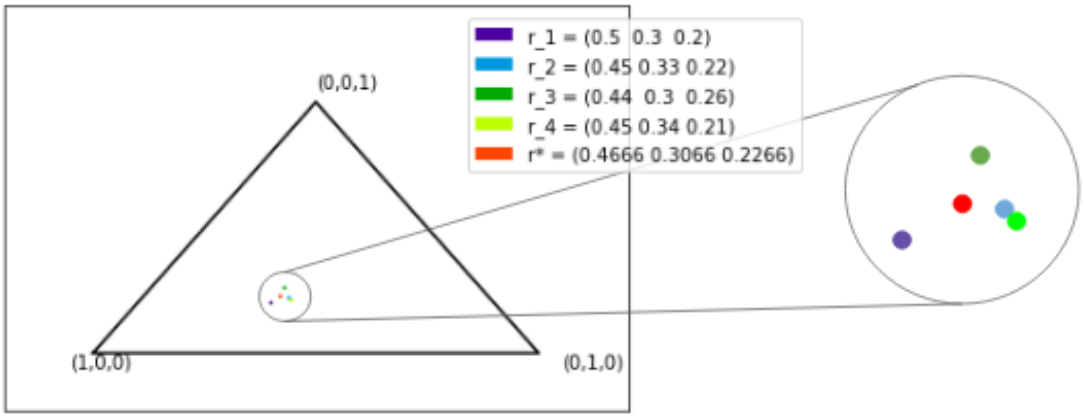
□

Example 7. Consider again the rows $r_1 = (0.5, 0.3, 0.2), r_2 = (0.45, 0.33, 0.22), r_3 = (0.44, 0.3, 0.26)$ and $r_4 = (0.45, 0.34, 0.21)$ from Lemma 2 that gave an empty $[u,v]_{opt}$, with representation given by $(0.47, [0.31, 0.33], 0.23)$. In this case, we have $0.47+0.31+0.23 > 1$, and

$$\delta = \frac{(0.47+0.31+0.23)-1}{3} = \frac{0.01}{3} = 0.003\dot{3}.$$

Hence let $r^* = (0.47 - 0.003\dot{3}, 0.31 - 0.003\dot{3}, 0.23 - 0.003\dot{3}) = (0.466\dot{6}, 0.306\dot{6}, 0.226\dot{6})$, with $\varepsilon^* = 0.033\dot{3}$. See Figure 2.3.

Figure 2.3: Example 7, when $[u, v]_{opt}$ is empty.



We have restricted our discussion so far to just producing a row that reduces the error for a single partition, which we assumed without loss of generality to be Q_1. One benefit of using transition sets to represent the set of vectors that optimally abstract the transition probabilities of each Q_i is that we can easily extend this to a family of transition probability matrices for the entire lumping that is either an optimal abstraction (in terms of reduced error) or close to optimal. Let us then return to our original DTMC $\mathcal{D} = (Q, q_0, P, L)$, with state space partition $Q_1, \ldots, Q_m$. We then have the following procedure for producing the corresponding interval Markov chain, $[\mathcal{D}] = (\mathcal{A}, Q_1, [\Pi], L)$, where $\mathcal{A} = \{Q_1, \ldots, Q_m\}$:

IMC Construction Procedure

For all $1 \leq i \leq m$ we construct the transition set, $[\Pi]$ of the IMC in the following way:

1. For $Q_i = \{q_1^i, \ldots, q_{k_i}^i\}$, we have vectors $r_1^i, \ldots, r_{k_i}^i$, where:
$$r_i^j = (P(q_j^i, Q_1), \ldots, P(q_j^i, Q_m)).$$
2. Let $\beta_i = \frac{1}{2} \max_{1 \leq a,b \leq k_i} \|r_a^i - r_b^i\|_\infty$.
3. For all $1 \leq j \leq m$, let $u_j^i = \min_{1 \leq a \leq k_i} r_{aj}^i$ and $v_j^i = \max_{1 \leq a \leq k_i} r_{aj}^i$.
4. Let $[u^i, v^i] = ([u_1^i, v_1^i], \ldots, [u_m^i, v_m^i])$.
5. So then let $[u^i, v^i]_{opt} = ([v_1^i - \beta_i, u_1^i + \beta_i], \ldots, [v_m^i - \beta_i, u_m^i + \beta_i])$.
6. Let $\delta_i = \max\{\frac{1 - \sum_{j=1}^m u_j^i}{m}, \frac{\sum_{j=1}^m v_j^i - 1}{m}\}$.

7. Let $[u^i, v^i]_{\delta_i} = ([v_1^i - \delta_i, u_1^i + \delta_i], \ldots, [v_m^i - \delta_i, u_m^i + \delta_i])$.

8. Let $R_i = \begin{cases} [u^i, v^i]_{opt} & \text{if } [u^i, v^i]_{opt} \text{ is non-empty,} \\ [u^i, v^i]_{\delta_i} & \text{otherwise.} \end{cases}$

Hence let $[\Pi]$ be the $m \times m$ matrix of intervals with rows $R_1, \ldots, R_m$. We have guaranteed by the construction of the R_i that $[\Pi]$ is non-empty, and hence $[\mathcal{D}] = (\mathcal{A}, Q_1, [\Pi], L)$ is a well-defined IMC.

2.3 Introducing the new approach

In the previous section, for the DTMC $\mathcal{D} = (Q, q_0, P, L)$ we produced a corresponding interval Markov chain $[\mathcal{D}] = (\mathcal{A}, Q_1, [\Pi], L)$ where a row of $[\Pi]$ consists of the vectors that have optimal (or close to optimal) error at the corresponding state. Recall that for IMCs there are two possible semantical interpretations of how the uncertainty is resolved. In the UMC semantics, a transition matrix $T \in [\Pi]$ is chosen by the environment and we must consider the DTMC (Q, q_0, T, L). Under the IMDP semantics, the environment makes a choice of distribution online at each state the model reaches. Under our interpretation, the uncertainty in $[\mathcal{D}]$ is introduced by us, and hence we can have adversarial control over the system, rather than assuming that the choose is made non-deterministically by the environment. This being the case, it prima facie seems more natural that we consider this new abstraction under the IMDP semantics. Furthermore, the above procedure also ensures that given any DTMC, we can construct a unique corresponding IMDP of optimally virtual points. The set of possible distributions we can choose from at each state are then the set of vectors with optimal error. We hence give our key definition:

Definition 38. Given a DTMC $\mathcal{D} = (Q, q_0, P, L)$, the *virtual Interval Markov Decision Process* (vIMDP) corresponding to $\mathcal{D}$ (with associated error β) is the unique IMDP constructed using the IMC Construction Procedure from $\mathcal{D}$.

We use the IMDP representation here as opposed to either the polyMDP or pMDP representations as our above procedure always produces a transition set of optimal vectors, which is what an IMC is represented by, and it leads to both compact notation, and brevity of construction; if for example we wanted to represent the construction as a polyMDP, we would have to further find the vertices of the corresponding polytopes as part of the representation.

Recall we stated the following theorem (Theorem 3) concerning model checking IMDPs: Given an IMDP $\mathcal{I} = (Q, q_0, \delta, L)$, there exists an MDP $\mathcal{M}$ such that for any PCTL formula ϕ, $\lceil \mathcal{M} \rceil \vDash \phi$ if and only if $\mathcal{M} \vDash \phi$.

If two models verify all and only the same PCTL formulae, we say there are PCTL-equivalent. We will demonstrate the method for constructing this further MDP in the following chapter on computational complexity, but we can hence define:

Definition 39. Given a DTMC $\mathcal{D} = (Q, q_0, P, L)$, a *virtual Markov Decision Process* (vMDP) corresponding to $\mathcal{D}$ (with associated error β) is an MDP that is PCTL-equivalent to the vIMDP corresponding to $\mathcal{D}$.

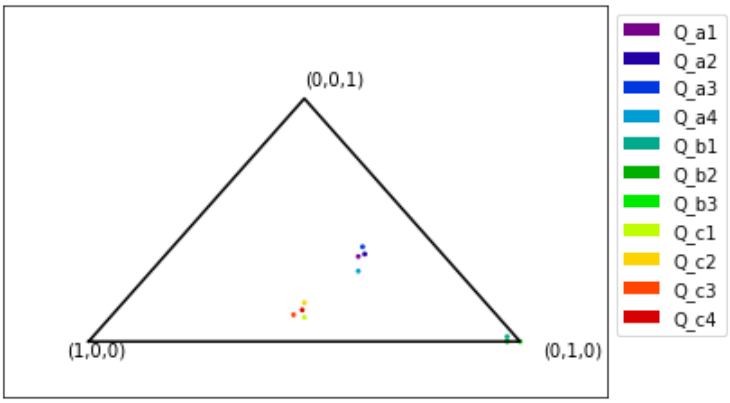
For a state of the vIMDP with actions among the transition set $[u, v]$, the corresponding state of the vMDP constructed has action set equal to the set of the vertices of $conv([u, v])$, and hence all the actions are still points with optimal error. We can hence perform model checking over this virtual MDP with the same assurance that the error at each state is still optimal in relation to the concrete model. We are interested in being able to determine how the probabilities of PCTL path formulae holding over the new abstraction compares to the probability of it holding over the concrete model. When considering whether a PCTL state formula of the form $\mathbb{P}_{\sim p}[\psi]$ is verified by an MDP at its initial state q_0, we need to calculate the values $p_{\min}(q_0, \psi)$ or $p_{\max}(q_0, \psi)$. We have introduced non-determinism into the abstracted model by considering it as an MDP, but, as was mentioned above, we can take adversarial control over the choice of actions at each state and hence always ensure that we choose an adversary that achieves either the maximum or minimum probabilities. Therefore, when comparing the probabilities of formulae holding over the new abstraction against the concrete model, we will choose adversaries to consider that obtain the maximum and minimum probabilities. We will return to this in Chapter 5.

Remark 3. Because we have adversarial control, we can assume a slight variation in PCTL semantics for interval Markov chains. Recall the following: given an IMC $\mathcal{I} = (Q, q_0, [\Pi], L)$, a PCTL formula ϕ, when considering the UMC semantics for IMCs, we have that $\mathcal{I} \vDash \phi$ if and only if for all $P \in [\Pi]$, the DTMC $\mathcal{D} = (Q, q_0, P, L)$ is such that $\mathcal{D} \vDash \phi$. Similarly recall that under the IMDP semantics, for $\tilde{\mathcal{I}} = (Q, q_0, \delta, L)$, $\tilde{\mathcal{I}} \vDash \phi$ if and only if for all adversaries, σ, at q_0, $\tilde{\mathcal{I}}^\sigma \vDash \phi$. As we have choice over the adversary, or the initial conditions of the IMDP, we now can say that $\tilde{\mathcal{I}} \vDash \phi$ if and only if for *any* adversary σ, at q_0, $\tilde{\mathcal{I}}^\sigma \vDash \phi$.

2.3.1 Case study

Here we create a small study on constructing the different abstractions above for model checking. We first demonstrate how a model is found by using the naive approach, then show how an abstracted vIMDP and vMDP are found by the new approach. We begin with our concrete model, an 11 state DTMC, $\mathcal{D} = (Q, q_0, P, L)$, with states $q_0, \ldots, q_{10}$ and

Figure 2.4: Rows given by the lumping of the DTMC.



transition probabilities given by the following matrix:

$$P = \begin{pmatrix} 0.05 & 0.05 & 0.05 & 0.05 & 0.15 & 0.15 & 0.15 & 0.3 & 0.02 & 0.01 & 0.02 \\ 0.04 & 0.04 & 0.05 & 0.05 & 0.14 & 0.17 & 0.15 & 0.28 & 0.03 & 0.03 & 0.02 \\ 0.01 & 0.01 & 0.1 & 0.05 & 0.14 & 0.15 & 0.15 & 0.2 & 0.19 & 0 & 0 \\ 0.06 & 0.04 & 0.06 & 0.07 & 0.16 & 0.17 & 0.15 & 0.07 & 0.03 & 0.05 & 0.14 \\ 0.01 & 0.01 & 0 & 0 & 0.96 & 0 & 0 & 0.005 & 0.005 & 0.005 & 0.005 \\ 0 & 0.01 & 0.01 & 0.01 & 0.01 & 0.95 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.25 & 0.5 & 0.25 & 0 & 0 & 0 & 0 \\ 0.15 & 0.15 & 0.15 & 0 & 0.15 & 0.15 & 0.15 & 0.02 & 0.03 & 0.03 & 0.02 \\ 0.15 & 0.15 & 0.06 & 0.06 & 0.14 & 0.13 & 0.15 & 0.04 & 0.03 & 0.03 & 0.06 \\ 0.4 & 0.04 & 0.02 & 0.01 & 0.14 & 0.13 & 0.15 & 0.1 & 0.01 & 0 & 0 \\ 0.44 & 0 & 0 & 0 & 0 & 0 & 0.43 & 0 & 0 & 0 & 0.13 \end{pmatrix}$$

and with the following labelling: $L(q_0) = L(q_1) = L(q_2) = L(q_3) = a$, $L(q_4) = L(q_5) = L(q_6) = b$, and $L(q_7) = L(q_8) = L(q_9) = L(q_{10}) = c$.

The labelling determines the lumping, so we have $Q_a = \{q_0, q_1, q_2, q_3\}$, $Q_b = \{q_4, q_5, q_6\}$, and $Q_c = \{q_7, q_8, q_9, q_{10}\}$. From this lumping we obtain the following rows corresponding to each element of each of Q_a, Q_b, Q_c:

Q_a: $(0.2, 0.45, 0.35), (0.18, 0.46, 0.36), (0.17, 0.44, 0.39), (0.23, 0.48, 0.29)$,

Q_b: $(0.02, 0.96, 0.02), (0.03, 0.97, 0), (0, 1, 0)$,

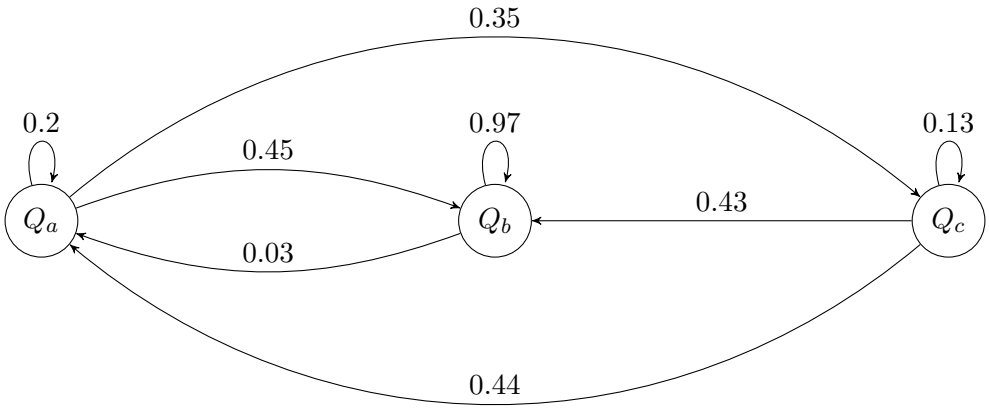
Q_c: $(0.45, 0.45, 0.1), (0.42, 0.42, 0.16), (0.47, 0.42, 0.11), (0.44, 0.43, 0.13)$.

Following the naive approach to producing a lumped abstraction by choosing the rows from each set above with the best error, we get a DTMC with transition probabilities given by the following matrix:

$$\begin{pmatrix} 0.2 & 0.45 & 0.35 \\ 0.03 & 0.97 & 0 \\ 0.44 & 0.43 & 0.13 \end{pmatrix}.$$

This DTMC is a 0.06-bisimulation of the concrete model, and the transition diagram is shown in Figure 2.5.

Figure 2.5: The DTMC obtained by the naive abstraction method.



We now follow the IMC Construction Procedure to produce the vIMDP corresponding to $\mathcal{D}$. We get the following optimal errors of any virtual points: $\beta_a = 0.05$, $\beta_b = 0.02$, $\beta_c = 0.03$ for the above collections of rows, respectively, and the transition sets of virtual points with optimal error for each collection of rows are $[u^a, v^a]_{opt} = ([0.18, 0.22], [0.43, 0.49], 0.34)$, $[u^b, v^b]_{opt} = ([0.01, 0.02], 0.98, [0, 0.02])$, and $[u^c, v^c]_{opt} = ([0.44, 0.45], [0.42, 0.45], 0.13)$, respectively, which are all non-empty.

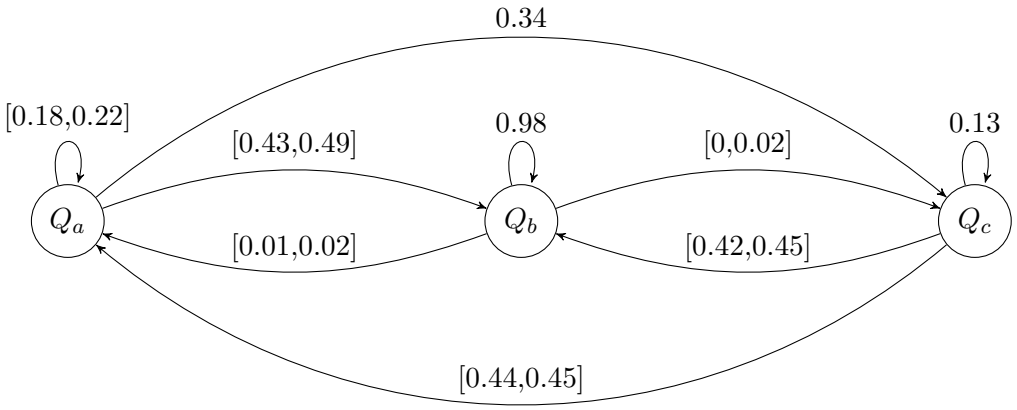
We hence obtain an IMC $\mathcal{I} = (\{Q_a, Q_b, Q_c\}, Q_a, [\Pi], L)$, where $[\Pi]$ is the transition set represented by the 3×3 matrix of intervals with rows $[u^a, v^a]_{opt}, [u^b, v^b]_{opt}, [u^c, v^c]_{opt}$, and $L(Q_a) = a, L(Q_b) = b, L(Q_c) = c$. We hence get the virtual IMDP:

$$\tilde{\mathcal{I}} = (\{Q_a, Q_b, Q_c\}, Q_a, \delta, L),$$

where δ is defined for $i = a, b, c$ as: $\delta(Q_i) = [u^i, v^i]_{opt}$. A transition diagram representation of this vIMDP is show in Figure 2.6.

The corresponding virtual MDP $\mathcal{M} = (\{Q_a, Q_b, Q_c\}, Q_a, \delta', L)$ is constructed from $\tilde{\mathcal{I}}$ by taking for each $i = a, b, c$, $\delta'(Q_i)$ to be the set of vertices of the convex hull of $\delta(Q_i)$ - a procedure for which can be found in [9], and will be discussed in more detail in the following chapter. Thus we get the following:

Figure 2.6: The virtual IMDP, $\mathcal{I}$, corresponding to the concrete model.



$$\delta'(Q_a) = \{\alpha = (0.18, 0.48, 0.34),\ \gamma = (0.22, 0.44, 0.34)\},$$

$$\delta'(Q_b) = \{\eta = (0.01, 0.98, 0.01),\ \zeta = (0.02, 0.98, 0)\},$$

$$\delta'(Q_c) = \{\chi = (0.44, 0.43, 0.13),\ \omega = (0.45, 0.42, 0.13)\}.$$

A transition diagram representation of this vMDP is show in Figure 2.7, and the actions available at state Q_a are seen shown in a geometrical representation in Figure 2.8.

2.4 A related work

In [18], interval Markov chains and bounded-parameter Markov decision processes are leveraged as abstractions of the following continuous-domain, discrete-time switched stochastic dynamical system:

$$x_{k+1} = \mathcal{F}_a(x_k, w_k), \quad x_k \in \mathcal{P} \subset \mathbb{R}^n,\ w_k \in \mathcal{W} \subset \mathbb{R}^n,$$

where:

- $a \in I_a = \{1, 2, \ldots, m_a\}$ a finite indexing set labelling the available dynamics of the system,
- $\mathcal{F}_a : \mathcal{P} \times \mathcal{W} \to \mathcal{P}$ is a possibly non-linear function,
- $\mathcal{P}$ is a full dimensional[1] polytope in $\mathbb{R}^n$,

[1] A convex polytope is full dimensional if it is an n-dimensional object in $\mathbb{R}^n$.

Figure 2.7: The MDP $\mathcal{M}$ equivalent to the virtual IMDP.

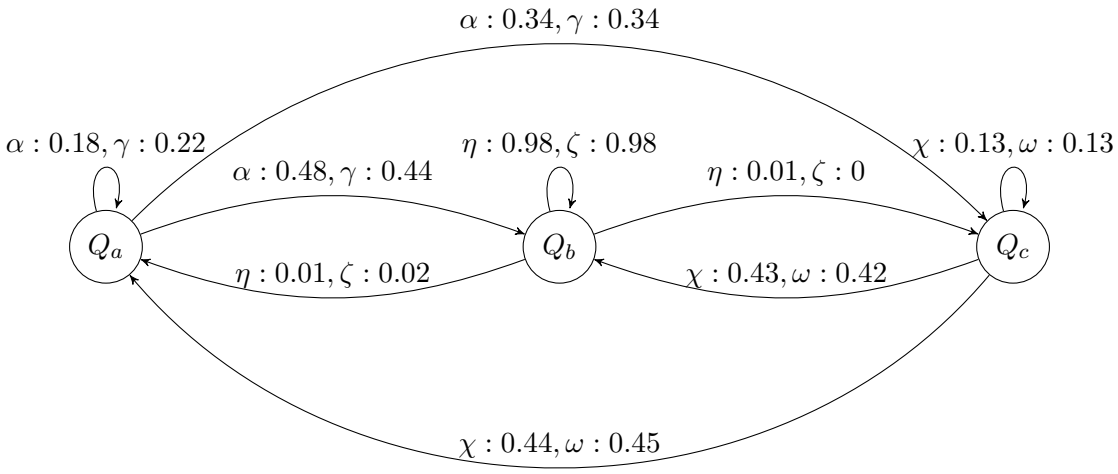
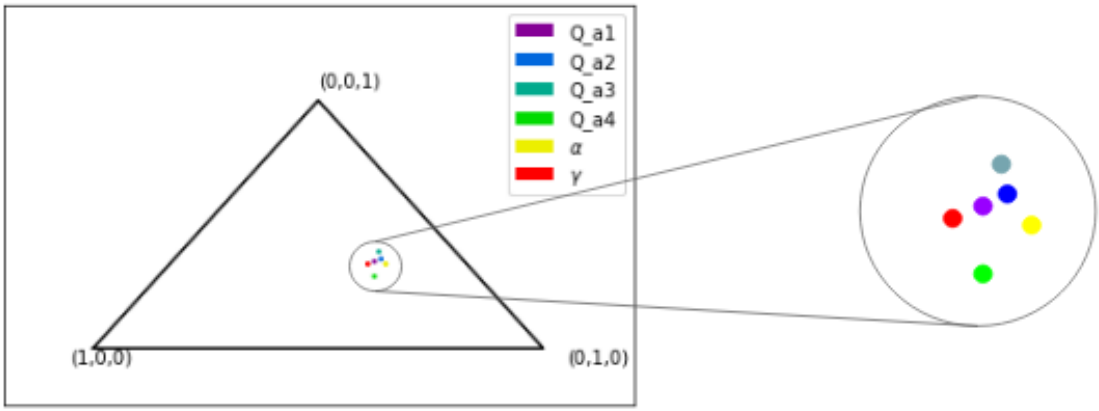


Figure 2.8: Actions of the virtual MDP at state Q_a. The possible actions at Q_a in the virtual IMDP are all the points on the straight line joining α and γ.



- $k \in \mathbb{N}$,
- w_k is a sample from a given probability distribution over a polyhedral subset $\mathcal{W} \subset \mathbb{R}^n$.

The goal in the work mentioned here is twofold: (1) to verify PCTL properties over this stochastic system, and (2) to synthesise adversaries that maximise the probability of the system satisfying PCTL properties. To tackle problem 1, this work outlines a construction procedure that produces an abstracted IMC in the case where $m_a = 1$, and a BMDP in the case where $m_a > 1$. For problem 2, m_a is assumed to be strictly greater than 1, and hence the system is abstracted to a BMDP. Model checking and synthesis is then performed over the abstractions, using the IMDP semantics for IMCs.

Chapter 3

Computational Complexity

We are interested in comparing the computational complexity of the received method of using approximate probabilistic bisimulation for the verification of PCTL properties, and using our new abstraction that makes use of virtual states.

Let us assume again that we are given a lumping of the state-space of a large DTMC and given a PCTL property ϕ. In the naive method, we first produce a smaller DTMC by taking a concrete representative from each partition (let us assume we take the concrete point with minimal error from each partition), and perform model checking of the property on the smaller state space DTMC. In the virtual abstraction method, we first reduce the large DTMC to an IMDP of optimal points (as outlined above), further convert the IMDP to a classical MDP, then perform model checking of the property on the resultant MDP. The relative computational complexities of model checking over DTMCs and MDPs is known, and here we will look at the relative complexity of converting the concrete model to both an abstracted DTMC, and to the virtual MDP.

3.1 Conversion to a virtual IMDP

Let us assume we are given a DTMC $\mathcal{D} = (Q, q_0, P, L)$, where $|Q| = n$, and given a partition of the state-space $\mathcal{A} = \{Q_1, \ldots, Q_m\}$ for some $m \leq n$. Let $k_1 = |Q_1|, \ldots, k_m = |Q_m|$. Let us focus on some Q_i for now, and determine the cost of the steps outlined above for this conversion.

1. For $Q_i = \{q^i_1, \ldots, q^i_{k_i}\}$, we have vectors $r^i_1, \ldots, r^i_{k_i}$, where for any $1 \leq j \leq k_i$, $r^i_j = (P(q^i_j, Q_1), \ldots, P(q^i_j, Q_m))$.

Each r^i_j takes at most m additions to calculate, so this step takes $\mathcal{O}(k_i m)$ time.

2. Let $\beta_i = \frac{1}{2} \max\limits_{1 \leq a,b \leq k_i} \|r^i_a - r^i_b\|_\infty$.

To calculate β_i requires comparing all pairs of r_a^i, r_b^i, and each comparison consists of m individual comparisons, so this step takes $\mathcal{O}(k_i^2 m)$ time.

3. For all $1 \leq j \leq m$, let $u_j^i = \min\limits_{1\leq a\leq k_i} r_{aj}^i$ and $v_j^i = \max\limits_{1\leq a\leq k_i} r_{aj}^i$.

4. Let $[u^i, v^i] = ([u_1^i, v_1^i], \ldots, [u_m^i, v_m^i])$.

Finding minima and maxima takes $\mathcal{O}(k_i m)$ time.

5. Then let $[u^i, v^i]_{opt} = ([v_1^i - \beta_i, u_1^i + \beta_i], \ldots, [v_m^i - \beta_i, u_m^i + \beta_i])$.

6. Check this is non-empty.

Step 5 takes $2m$ calculations, and step 6 involves checking that $\sum_{j=1}^m (v_j^i - \beta_i) \leq 1$ and $\sum_{j=1}^m (u_j^i + \beta_i) \geq 1$, so these steps together take $\mathcal{O}(m)$ time. We then let $R_i = [u^i, v^i]_{opt}$. If $[u^i, v^i]_{opt}$ is empty we additionally have the following steps:

(**7.**) Let $\delta_i = \max\{\frac{1-\sum_{j=1}^m u_j^i}{m}, \frac{\sum_{j=1}^m v_j^i - 1}{m}\}$.

(**8.**) Let $[u^i, v^i]_{\delta_i} = ([v_1^i - \delta_i, u_1^i + \delta_i], \ldots, [v_m^i - \delta_i, u_m^i + \delta_i])$.

These similarly together would take $\mathcal{O}(m)$ time. If these steps are required, we then let $R_i = [u^i, v^i]_{\delta_i}$. We have now constructed a row of the IMC. Now note that it may be the case that $k_i > m$. Steps **1-4** together take $\mathcal{O}(k_i^2 m)$ time and hence we can say that the whole procedure for any arbitrary Q_j takes $\mathcal{O}(\max\limits_{1\leq i\leq m} k_i^2 m)$. Hence for all m rows of the IMC to be produced it takes $\mathcal{O}(\max\limits_{1\leq i\leq m} k_i^2 m^2)$ time.

3.2 Conversion to a lumped Markov chain

The process of converting the same concrete model given above to a lumped DTMC in contrast only requires the following two steps:

- For $Q_i = \{q_1^i, \ldots, q_{k_i}^i\}$, we have $r_1^i, \ldots, r_{k_i}^i$, where $r_j^i = (P(q_j^i, Q_1), \ldots, P(q_j^i, Q_m))$.

Again, each r_j^i takes at most m additions to compute, so this step takes $\mathcal{O}(k_i m)$ time.

- Let $R_i = \operatorname{argmin}\limits_{1\leq a\leq k_i} \left(\max\limits_{1\leq b\leq k_i} \|r_a^i - r_b^i\|_\infty \right)$.

To find R_i requires comparing all pairs of r_a^i, r_b^i, and each comparison consists of m individual comparisons, so this step takes $\mathcal{O}(k_i^2 m)$ time. We must repeat these two steps for all $Q_1, \ldots, Q_m$, and thus the whole process of finding the lumped DTMC also takes $\mathcal{O}(\max\limits_{1\leq i\leq m} k_i^2 m^2)$ time. Thus the computational cost of finding the IMC related to the lumping and the DTMC related to the lumping is of the same order.

3.3 Conversion to an equivalent virtual MDP

Now that we have considered conversion from the concrete model to the two candidate lumped models, we should compare the cost of performing model checking over these constructions. Recall that in our case study for our IMDP model we converted the IMDP to its corresponding MDP, and then performed standard MDP model checking. This was suitable for such a small model, but we will see in the following that this process is exponentially costly. Given a PTCL formula, ϕ and a DTMC (Q, q_0, P, L), we know that the model checking complexity is linear in the size of ϕ (the number of logical connectives and temporal operators plus the size of the bounds on the bounded-time operators) and polynomial in $|Q|$ ([1],[19]). Similarly the complexity of model checking the same formula over an MDP (Q, q_0, δ, L) is also linear in $|\phi|$ and polynomial in $|Q|$, with a finite constant factor of the order of the number of actions [1].

The following two steps (9. and 10., continuing from above) outline the conversion of this IMDP row to a "row" of an MDP, i.e. the actions we consider at the state Q_i. R_i is a transition set, and our goal now is to find the vertices of the polytope that R_i describes. This is achieved in two steps, following methods outlined in [9]. An equivalent method is also given in [11], using the "basic feasible solutions" to the set of inequalities described by R_i.

The first step is to tighten the intervals in R_i. This is a necessary step in order to run the vertex finding algorithm (below) to produce the vertices of the polytope described by R_i.

Definition 40. Let $[u, v]$ be a $1 \times m$ transition set. If for any $1 \leq i \leq m$, $u_i = \min_{x \in [u,v]} x_i$ and $v_i = \max_{x \in [u,v]} x_i$, then u_i and v_i are *tight*, respectively. If for all $1 \leq i \leq m$, u_i and v_i are tight, then we also say that $[u, v]$ is tight.

Any such transition set, if not tight, can be tightened; i.e. for any $[u, v]$, there is tight $[\bar{u}, \bar{v}]$ such that $[u, v] = [\bar{u}, \bar{v}]$. Such a tightened interval is found by the following procedure:

Tight Interval Algorithm.

1. Input u, v, vectors of size l.
2. For all $1 \leq i \leq l$, do
   (a) If $u_i + \sum_{k \neq i} v_k \geq 1$, set $\bar{u_i} = u_i$. Else, set $\bar{u_i} = 1 - \sum_{k \neq i} v_k$.
   (b) If $v_i + \sum_{k \neq i} u_k \leq 1$, set $\bar{v_i} = v_i$. Else, set $\bar{v_i} = 1 - \sum_{k \neq i} u_k$.
3. Output $\bar{u}, \bar{v}$.

Stop.

This algorithm runs in time $\mathcal{O}(l^2)$. We hence have as our next step in our conversion procedure:

9. Tighten R_i.

which we now know will run in time $\mathcal{O}(m^2)$. In what follows we will write $\bar{R}_i = [a, b]$ for the tightened representation of R_i. To find the corners of $\bar{R}_i$, we need the following definition and lemma:

Definition 41. Let $[u, v]$ be a tight $1 \times m$ transition set, and $x \in [u, v]$. We say that the element x_i of x is *free* if $u_i < x_i < v_i$.

Lemma 3. Let $[u, v]$ be a tight $1 \times m$ transition set. A vector $x \in [u, v]$ is a vertex of $[u, v]$ if and only if x has at most one free element.

It follows from Lemma 3 that any tight $1 \times m$ transition set can have at most $m2^{m-1}$ vertices. An example of a candidate vertex would be $(u_1, u_2, \ldots, u_{m-1}, 1-\sum_{j\neq m} u_i)$, which is a vertex if $u_m \leq 1 - \sum_{j\neq m} u_i \leq v_m$. The following procedure for finding all the vertices uses this lemma, which is why we must first ensure that we have tightened R_i:

Vertex Finding Algorithm.

1. Input $[u, v]$, a tight $1 \times l$ transition set. Set $\mathcal{V} = \emptyset$. Set $\mathcal{U} = \{y \in [u, v] \mid \forall 1 \leq i \leq l, x_i \in \{u_i, v_i\}\}$
2. For all $1 \leq i \leq l$, do
   (a) For all $y \in \mathcal{U}$, do
      (i) Set $y_i = 1 - \sum_{k\neq i} y_k$.
      (ii) If $u_i \leq y_i \leq v_i$, set $\mathcal{V} = \mathcal{V} \cup \{y\}$.
3. Output $\mathcal{V}$.

Stop.

This algorithm runs in time $\mathcal{O}(l^2 2^{l-1})$. Our final step of the conversion is to determine the vertices (actions of the MDP) corresponding to $\bar{R}_i$:

10. Let $\mathcal{V}_i$ be the vertices of the polytope described by $\bar{R}_i$.

We know that finding $\mathcal{V}_i$ takes $\mathcal{O}(m^2 2^{m-1})$ time. Letting $\mathcal{M} = (\mathcal{A}, Q_1, \delta, L)$, where for any $Q_i \in \mathcal{A}$, $\delta(Q_i) = \mathcal{V}_i$, we hence find that the addition procedure of converting the IMDP to its corresponding MDP, $\mathcal{M}$ takes $\mathcal{O}(m^3 2^{m-1})$ time by this method. Therefore the entire process of obtaining the virtual MDP from the concrete model takes $\mathcal{O}(m^3 2^{m-1} + \max_{1\leq i\leq m} k_i^2 m^2)$ time. Comparing this to the time it takes to obtain the lumped DTMC, $\mathcal{O}(\max_{1\leq i\leq m} k_i^2 m^2)$, we see that there is a significant increase in complexity. Furthermore, as this method produces an MDP with exponentially many actions (exponential in the the size of the state space, m), model checking over this MDP would also be exponential in the size of the state space. In the following section we see how we can navigate around this complexity explosion.

3.4 Polynomial time model checking of IMDPs

In [7], a polynomial time verification procedure for PCTL properties over IMDPs is given. The proof is given for CMDPs, but recall that IMDPs, and furthermore our virtual IMDP constructed above, fall under the umbrella of CMDPs. First we must determine the size of a CMDP, $\mathcal{C} = (Q, q_0, \gamma, L)$. Recall that for all $s \in Q$, $\gamma(s)$ is a finite set $\{C_s^1, \ldots C_s^{K_s}\}$, such that for all $1 \leq i \leq K_s$, C_s^i is a non-empty convex set of stochastic vectors. Let $N = |Q|$ and $M = \max_{s \in Q} |\gamma(s)|$, the maximum number of actions at a state. Hence the maximum possible number of transitions of the model is $\mathcal{O}(N^2 M)$. For each $s \in Q$ and each $C_s^i \in \gamma(s)$, let D_s^i denote the number of constraints required to express the convex uncertainty set C_s^i. Let $D = \max_{s \in Q} \max_{C_s^i \in \gamma(s)} D_s^i$. So the amount of data required to express the function γ fully is $\mathcal{O}(NMD)$. Therefore we get that $|\mathcal{C}| = \mathcal{O}(N^2 M + NMD)$.

We then have the following theorem:

Theorem 4. Given a CMDP $\mathcal{C} = (Q, q_0, \gamma, L)$:

1. The problem of verifying whether $\mathcal{C}$ of size N satisfies a PCTL formula ϕ not containing any bounded until operators is in PTIME.

2. A formula ϕ' containing bounded until operators can be verified with time complexity $\mathcal{O}(poly(|\mathcal{C}|) \times |\phi'| \times k_{max})$, where k_{max} is the maximum time bound on any bounded-until operator in ϕ'.

We give a very cursory sketch of the verification algorithm constructed that proves this theorem. The formula ϕ is parsed in time linear in $|\phi|$. The satisfiability set of each operator is then computed. For non-probabilistic boolean operators, this is done in time polynomial in $|\mathcal{C}|$ in the standard way (see [1] for verification procedures of MDPs). For the probabilistic operators, the satisfiability sets are computed by generating and then solving a number of convex optimisation problems, of size polynomial in $|\mathcal{C}|$. For full details of these convex problems, see [7].

This theorem holds under a number of assumptions about the model. All but one of these assumptions hold for all IMDPs, so we will only bring attention to the odd one out. It is the following:

Assumption 1. Given a $\mathcal{C} = (Q, q_0, \gamma, L)$, for all $s \in Q$, for all $C \in \gamma(s)$, if a transition probability is zero in some $v_0 \in C$, then that transition probability is zero for all $v \in C$. Formally: $\forall s \in Q,\ \exists v \in C : v_s = 0 \Rightarrow \forall v \in C : v_s = 0$.

This assumption guarantees the correctness of the verification procedures, which rely on state reachability. If for an action at state s, the probability of traversing to state s' was sometimes zero and sometimes not zero, sometimes s' would be reachable along that

action, and sometimes it wouldn't, which would affect the correctness of the reachability analysis.

As an example, recall the IMDP constructed in our case study, with transition set given by:

$$\begin{pmatrix} [0.18, 0.22] & [0.43, 0.49] & 0.34 \\ [0.01, 0.02] & 0.98 & [0, 0.02] \\ [0.44, 0.45] & [0.42, 0.45] & 0.13 \end{pmatrix}.$$

The middle row of this transition set admits the vectors $(0.02, 0.98, 0)$ and $(0.01, 0.98, 0.01)$, hence Assumption 1 does not hold for this model.

This assumption is needed for models where the transition probabilities at each state are chosen non-deterministically by the environment. However, recalling Remark 3, we are the agent who selects the action at each state, and hence properties need not be evaluated for all distributions within the convex set of possible actions. Thus in our case, Assumption 1 can be dropped. We just need not consider the action $(0.02, 0.98, 0)$ as a possible action in the above vIMDP.

We can determine the size of the virtual IMDP constructed from the DTMC $\mathcal{D} = (Q, q_0, P, L)$, which we will denote $\mathcal{I}$. Written as a CMDP, we have $\mathcal{I} = (\mathcal{A}, Q_1, \delta, L)$, where for all $Q_i \in \mathcal{A}$, $|\delta(Q_i)| = 1$ and is the set containing just the row R_i. In the above formulation of the size of a CMDP above, we now get that $N = |\mathcal{A}| = m$, and $M = 1$. Each row R_i consists of at most m intervals, and hence the maximum number of constraints to express the convex uncertainty is $D = 2m$. Plugging into the above formulation, we hence get that $|\mathcal{I}| = \mathcal{O}(m^2 + 2m^2) = \mathcal{O}(m^2)$. It follows that the algorithm for model checking our virtual IMDPs on the whole of PCTL (including bounded-until) runs in time $\mathcal{O}(poly(m^2) \times |\phi| \times k_{max})$.

We briefly mention a corollary to Theorem 4. As MDPs are a special case of CMDPs (without uncertainties) and DTMCs are special cases of MDP (one action at each state), Theorem 4 can be applied to them also. In the case of an MDP with N states and with a maximum M actions at each state, the size is $\mathcal{O}(N^2M)$, and hence the algorithm runs in time $\mathcal{O}(poly(N^2M) \times |\phi| \times k_{max})$. In the case of a DTMC with N states, we hence get that the algorithm runs in time $\mathcal{O}(poly(N^2) \times |\phi| \times k_{max})$. Assumption 1 trivially holds for MDPs and DTMCs, as the convex uncertainty sets only ever contain one point in these models.

3.5 Summary

In this chapter we have established bounds on the computational complexity of performing model checking over the new approach. We have followed techniques from [9] and the result from [11] to study the complexity of model checking via conversion to an equivalent virtual MDP, and seen that this implies exponentially-bounded running time, a substantial failing when compared to the complexity of model checking using the naive approach. However,

using work from [7], and exploiting their approach via convex optimisation, it seems that there are promising polynomial-time bounds that have been established on model checking IMDPs. This indicates that the procedure of model checking via the new approach is computationally feasible, and comparable to existing methods in terms of computing time. As one of the primary reasons to use such abstractions is due to the empirical computational intractability of running model checking algorithms over concrete model with a very large state space, this is an essential result for our project.

Chapter 4

A Geometric Account

In this chapter, we give an intuitive geometric account of the steps involved in producing the virtual IMDP corresponding to a DTMC with a given lumping. We aim to develop some insight into the shape that convex hull of the virtual points can take.

In the above procedure, for each A_i, we have a number of "concrete" rows, $r^i_1, \ldots, r^i_{k_i}$, and a "virtual" row of intervals, R_i. Here we are interested in looking at the shape of R_i – which as we know from our analysis of uncertain Markov chains is a convex polytope – in relation to the convex polytope defined by the "concrete" rows, $conv(\{r^i_1, \ldots, r^i_{k_i}\})$. In Examples 5 and 6, we saw that there are cases where $R_i \subseteq conv(\{r^i_1, \ldots, r^i_{k_i}\})$, and cases where this doesn't hold, respectively. In this section we look at whether we can characterise these cases. For notational simplicity, we will write our concrete rows as $r_1, \ldots, r_k$, and the virtual row as $R = ([v_1 - \beta, u_1 + \beta], \ldots, [v_m - \beta, u_m + \beta])$ $(= [u, v]_{opt})$, where $\beta = \frac{1}{2} \max_{1 \leq a,b \leq k} \|r_a - r_b\|_\infty$.

We clearly have the following proposition:

Proposition 10. For a set of stochastic vectors $r_1, \ldots, r_k$ the transition set $[u, v]_{opt}$ corresponding to these vectors can only have cardinality among $\{0, 1, 2^{\aleph_0}\}$.[1]

Proof. We know by Example 5, that $[u, v]_{opt}$ can be empty. If it is non-empty, we know it is a convex polytope in $\mathbb{R}^m$, and thus we know that it must either have cardinality 1 or $2^{\aleph_0}$. □

The vectors $r_1, \ldots, r_k$ all lie in $\mathbb{R}^m$. More specifically, there are members of the set of m-dimension stochastic vectors, which is a regular $(m-1)$-simplex. We will denote this $\mathbf{1}^m$, and we see that in $\mathbb{R}^2$ it is a line segment, in $\mathbb{R}^3$ a bounded plane, and so on. Let us take the data of R, i.e. the intervals, and define $\tilde{R} = \{t \in \mathbb{R}^m \mid \forall 1 \leq i \leq m,\ v_i - \beta \leq t_i \leq u_i - \beta\}$. $\tilde{R}$ is an m-dimensional hypercube, and $R = \tilde{R} \cap \mathbf{1}^m$. We can hence gain some intuition about the above three cases for the size of R in terms of this picture.

[1] $\aleph_0$ is the cardinality of the natural numbers, and $2^{\aleph_0}$ is the cardinality of the real numbers.

We firstly know that R is non-empty if and only if:

$$\sum_{i=1}^{m}(v_i - \beta) \leq 1 \leq \sum_{i=1}^{m}(u_i + \beta) \;\Leftrightarrow\; \sum_{i=1}^{m} v_i - m\beta \leq 1 \leq \sum_{i=1}^{m} u_i + m\beta$$

Intuitively, this is if some point of the hypercube lies on one side of the simplex, and some point lies on the other side, thus giving them non-empty intersection.

For each $1 \leq i \leq m$, let $\beta_i = \frac{v_i - u_i}{2}$. We can characterise the above set of inequalities in terms of the individual ranges of values for each element of the vectors $r_1, \ldots, r_k$. So we have that $v_i = u_i + 2\beta_i$, and we get:

$$\sum_{i=1}^{k}(v_i - \beta) = \sum_{i=1}^{m}(u_i + 2\beta_i - \beta).$$

Letting $\eta_i = \beta - \beta_i$ more generally we get R is non-empty iff:

$$\begin{aligned}
&\sum_{i=1}^{m}(u_i + 2\beta_i - \beta) \leq 1 \leq \sum_{i=1}^{m}(u_i + \beta) \\
&\Leftrightarrow \sum_{i=1}^{m}(u_i + \beta_i - \eta_i) \leq 1 \leq \sum_{i=1}^{m}(u_i + \beta_i + \eta_i) \\
&\Leftrightarrow \sum_{i=1}^{m}(u_i + \beta_i) - \sum_{i=1}^{m}\eta_i \leq 1 \leq \sum_{i=1}^{m}(u_i + \beta_i) + \sum_{i=1}^{m}\eta_i \\
&\Leftrightarrow \left|1 - \sum_{i=1}^{m}(u_i + \beta_i)\right| \leq \sum_{i=1}^{m}\eta_i.
\end{aligned}$$

4.1 $|R| = 1$

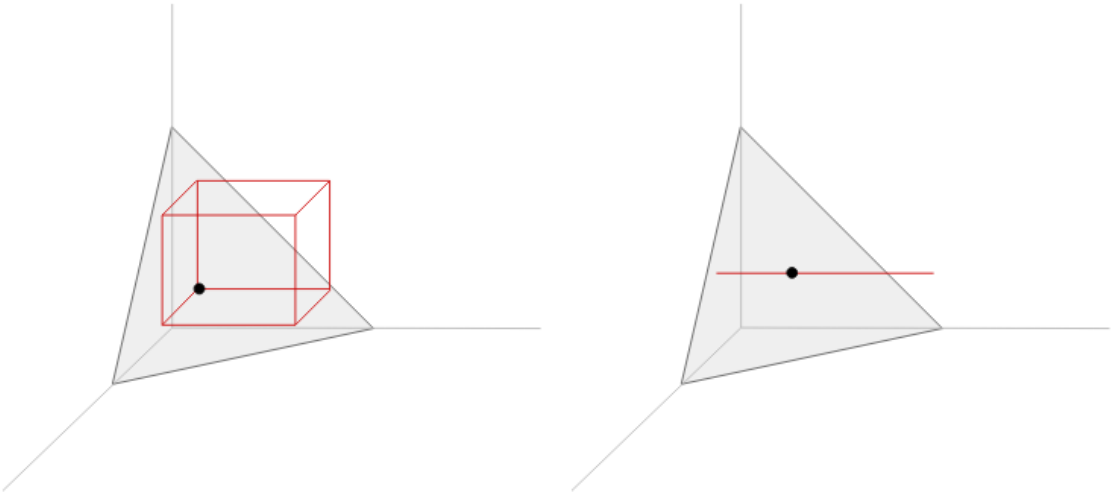
In the case that $|R| = 1$, there are two possibilities: either a single corner of the hypercube is touching the simplex, or the hypercube is in fact a line that intersects the simplex, as in Figure 4.1.

We can characterise the cases where $|R| = 1$ exactly as follows:

Lemma 4. Let $T = ([t_1, w_1], \ldots, [t_m, w_m])$ be any transition set. Then $|T| = 1$ if and only if at least one of the following conditions holds:

1. $\sum_{i=1}^{m} t_i = 1$
2. $\sum_{i=1}^{m} w_i = 1$
3. $\sum_{i=1}^{m} t_i < 1 < \sum_{i=1}^{m} w_i$ and T has only one free element.

Figure 4.1: The cases in $\mathbb{R}^3$ in which a hypercube has size 1 intersection with the regular 2-simplex.



Proof. For the backwards direction, suppose that $\sum_{i=1}^{m} t_i = 1$. Then $t = (t_1, \ldots, t_m) \in T$. If $a = (a_1, \ldots, a_m) \in T$, where $a \neq t$, then for all $1 \leq i \leq m$, $t_i \leq a_i$, and for some $1 \leq j \leq m$, $a_j > t_j$. This implies that $\sum_{i=1}^{m} a_i > 1$, and hence $a \notin T$, a contradiction. Thus $a = t$, and hence $T = \{t\}$. So $|T| = 1$. The argument for $\sum_{i=1}^{m} w_i = 1$ is similar.

Suppose that $\sum_{i=1}^{m} t_i \leq 1 \leq \sum_{i=1}^{m} w_i$ and T has only one free element. WLOG, we assume that the first element of T is free, so $t_1 < w_1$, and for all $1 < i \leq m$, $t_i = w_i$. Then $t_1 < 1 - \sum_{i \neq 1} t_i = 1 - \sum_{i \neq 1} w_i < w_1$, and hence $(1 - \sum_{i \neq 1} t_i, t_2, \ldots, t_m) \in T$. And clearly, as the first is the only free element of T, $T = \{(1 - \sum_{i \neq 1} t_i, t_2, \ldots, t_m)\}$.

For the forwards direction, we go by contraposition, assuming that none of the conditions hold. If either $\sum_{i=1}^{m} t_i > 1$ or $\sum_{i=1}^{m} w_i < 1$ we know that T must be empty, so we must suppose that $\sum_{i=1}^{m} t_i < 1 < \sum_{i=1}^{m} w_i$. In which case, we must further assume by the third condition that T has at least 2 free elements. Suppose WLOG that the first k elements are free, so $T = ([t_1, w_1], \ldots, [t_k, w_k], t_{k+1}, \ldots, t_m)$. For any $a = (a_1, \ldots, a_m) \in T$ there is at least one $1 \leq i \leq k$ such that $a_i > t_i$, and at least one $1 \leq j \leq k$ such that $a_j > w_i$. So suppose WLOG that $a_1 > t_1$ and $a_2 < w_2$. Let $\varepsilon = \min\{a_1 - t_1, w_2 - a_2\}$. Then also $a' = (a_1 - \varepsilon, a_2 + \varepsilon, a_3, \ldots, a_m) \in T$, and hence $|T| > 1$. □

We can hence apply Lemma 4 to $R = ([v_1 - \beta, u_1 + \beta], \ldots, [v_m - \beta, u_m + \beta])$ to determine exactly the cases in which $|R| = 1$.

Returning to the above intuitive geometric account of R, we can see that conditions (1) and (2) of the lemma are cases when the hypercube just touches the simplex with one of its vertices, and condition (3) is the case in which the hypercube is a line which intersects

with the simplex.

If $|R| = 1$, is it always the case that $R \subseteq conv(\{r_1, \ldots, r_k\})$? Let us examine this by the 3 cases in Lemma 4. First an example showing that when condition (1) of the lemma holds, the point can lie outside the polytope:

Example 8. Let us consider rows $r_1 = (0.3, 0.2, 0.2, 0.3)$, $r_2 = (0.2, 0.3, 0.2, 0.3)$, and $r_3 = (0.2, 0.2, 0.3, 0.3)$. We get $[u, v] = ([0.2, 0.3], [0.2, 0.3], [0.2, 0.3], 0.3)$, and $\beta = 0.05$, so $R = (0.25, 0.25, 0.25, [0.25, 0.35])$. Note that $0.25 + 0.25 + 0.25 + 0.25 = 1$, so this is an example of condition (1) from the lemma, and hence $R = \{(0.25, 0.25, 0.25, 0.25)\}$. Furthermore, we can see that $(0.25, 0.25, 0.25, 0.25)$ does not lie in $conv(\{r_1, r_2, r_3\})$.

A symmetric example can be given to show that when condition (2) of the lemma holds the point can lie outside the polytope:

Example 9. Let us consider rows $r_1 = (0.2, 0.3, 0.3, 0.2)$, $r_2 = (0.3, 0.2, 0.3, 0.2)$, and $r_3 = (0.3, 0.3, 0.2, 0.2)$. We get $[u, v] = ([0.2, 0.3], [0.2, 0.3], [0.2, 0.3], 0.2)$, and $\beta = 0.05$, so $R = (0.25, 0.25, 0.25, [0.15, 0.25])$. Again $0.25+0.25+0.25+0.25 = 1$, so this is an example of condition (2) from the lemma, and hence $R = \{(0.25, 0.25, 0.25, 0.25)\}$. Furthermore, we can see again that $(0.25, 0.25, 0.25, 0.25)$ does not lie in $conv(\{r_1, r_2, r_3\})$.

Finally we see that when condition (3) is satisfied, the point may still lay outside the polytope:

Example 10. Consider the rows $r_1 = (0.2, 0.2, 0.2, 0.2, 0.2)$, $r_2 = (0.1, 0.3, 0.18, 0.22, 0.2)$, and $r_3 = (0.15, 0.25, 0.1, 0.3, 0.2)$. We have that $\beta = 0.05$, and hence it follows that $R = (0.15, 0.25, 0.15, 0.25, [0.15, 0.25])$ in this case, thus it satisfies condition (3) of the lemma. This consists of the single point $r^* = (0.15, 0.25, 0.15, 0.25, 0.2)$. To show r^* does not lie within $conv(\{r_1, r_2, r_3\})$, suppose for contradiction that there are non-negative λ_1, λ_2, λ_3 such that $\sum \lambda_i = 1$ and $r^* = \sum \lambda_i r_i$. Then consider the first elements of each vector: we must have $0.2\lambda_1 + 0.1\lambda_2 + 0.15\lambda_3 = 0.15$. There are 3 possible solutions to this:

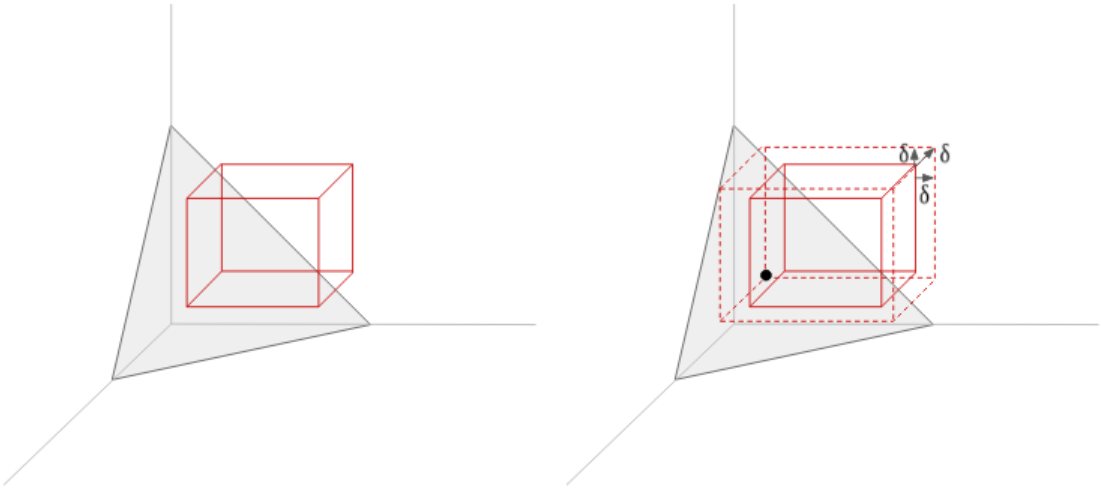
- $\lambda_1 = \frac{1}{3}, \lambda_2 = \frac{1}{3}, \lambda_3 = \frac{1}{3}$
- $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 1$
- $\lambda_1 = \frac{1}{2}, \lambda_2 = \frac{1}{2}, \lambda_3 = 0$

Considering now the third element of each vector, we must have that $0.2\lambda_1 + 0.18\lambda_2 + 0.1\lambda_3 = 0.15$, and none of the above solutions also satisfy this. Hence there are no such λ_i, and $r^* \notin conv(\{r_1, r_2, r_3\})$.

4.2 R is empty

In the case that $|R| = 0$, this means that $\tilde{R} \cap \mathbf{1}^m = \emptyset$. In this case, recall we loosen the intervals. This process is equivalent to uniformly expanding the hypercube $\tilde{R}$ in all directions until it just touches $\mathbf{1}^m$.

Figure 4.2: The case in $\mathbb{R}^3$ in which a hypercube has empty intersection with the 2-simplex, and its uniform expansion to the point where it touches the 2-simplex.



We loosen the intervals defining R by replacing β in the definition of R with $\delta = \max\{\frac{1-\sum_{i=1}^{m} u_i}{m}, \frac{\sum_{i=1}^{m} v_i - 1}{m}\}$. In this case, we are guaranteed that the transition set $R' = ([v_1 - \delta, u_1 + \delta], \ldots, [v_m - \delta, u_m + \delta])$ $(= [u, v]_\delta)$ is non-empty, and can also, as we saw in Example 7, still provide an improvement on the error achievable by the naive approach. Suppose that $\delta = \frac{\sum_{i=1}^{m} v_i - 1}{m}$. Then:

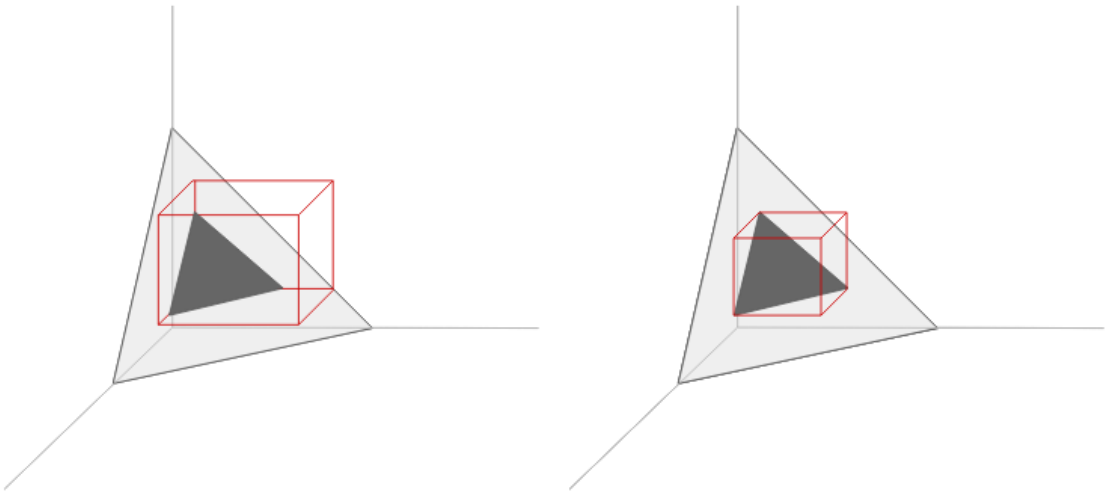
$$\sum_{i=1}^{m}(v_i - \delta) = \sum_{i=1}^{m} v_i - m\delta = \sum_{i=1}^{m} v_i - (\sum_{i=1}^{m} v_i - 1) = 1.$$

Hence, by Lemma 4, $|R'| = 1$. The result is the same if we instead let $\delta = \frac{1-\sum_{i=1}^{m} u_i}{m}$. We expected this result from the construction of δ. We intended to loosen the intervals "just enough" to ensure non-emptiness, or in terms of our geometric account, until the hypercube just touches the simplex, as seen in Figure 4.2.

4.3 $|R| = 2^{\aleph_0}$

In the case that $|R| = 2^{\aleph_0}$, there is a single possibility; the hypercube completely intersects (i.e. not just at a point) the simplex. When this happens and we perform the Tightening Algorithm, we are determining the smallest (in terms of volume) hypercube T such that $T \cap \mathbf{1}^m = \tilde{R} \cap \mathbf{1}^m$. In which case the vertices of the polytope $T \cap \mathbf{1}^m$ will be a subset of the vertices of the hypercube T. A picture of this in $\mathbb{R}^3$ is given in Figure 4.3.

Figure 4.3: The case in $\mathbb{R}^3$ in which a hypercube has infinite intersection with the 2-simplex, and the smallest hypercube with the same intersection.



We have from the previous sections necessary and sufficient conditions for the size of R being 0 or 1, and hence by Proposition 10 we have necessary and sufficient conditions for $|R| = 2^{\aleph_0}$. There are 3 cases we could characterise:

(i) $R \subseteq conv(\{r_1^i, \ldots, r_{k_i}^i\})$,

(ii) $R \cap conv(\{r_1^i, \ldots, r_{k_i}^i\}) = \emptyset$,

(iii) $R \cap conv(\{r_1^i, \ldots, r_{k_i}^i\}) \neq \emptyset$, but $R \nsubseteq conv(\{r_1^i, \ldots, r_{k_i}^i\})$.

Consider the transition set that has as its elements the interval ranges of the corresponding elements of concrete rows, i.e. $[\Pi] = ([u_1, v_1], \ldots, [u_m, v_m])$. Define $[\tilde{\Pi}] = \{t \in \mathbb{R}^m \mid \forall 1 \leq i \leq m,\ u_i \leq t_i \leq v_i\}$. $[\tilde{\Pi}]$ is the smallest hypercube in $\mathbb{R}^m$ that contains all of the vectors $r_1, \ldots, r_k$, with the i-th dimension of the hypercube given by the values $(v_i - u_i)$. This hypercube will most likely contain stochastic vectors that do not lie in $conv(\{r_1, \ldots, r_k\})$ – most exceptions will likely arise in pathological cases where $\tilde{\Pi} \cap \mathbf{1}^m = conv(\{r_1, \ldots, r_k\})$. We then shrink or expand the dimensions using the value β, obtaining a hypercube with dimensions given by $((u_i + \beta) - (v_i - \beta)) = (2\beta - (v_i - u_i))$. If $(v_i - u_i) < \beta$, then the i-th dimension of the hypercube increases, otherwise it decreases. At least one dimension always becomes 0, as there is always an interval of length 2β, by definition of β. If some of the dimensions of the hypercube increase, we would expect there to be some stochastic vectors falling within the resulting transition set that fall outside of the convex hull of the r_i, just as we expect there to be some such vectors in $\tilde{\Pi}$. If all of

the values decrease, depending on the shape of $conv(\{r_1, \ldots, r_k\})$ there will be cases where all the vectors lie in the convex hull, and cases where some (or all) do not, depending on the shape of the convex hull. There will be odd cases where there are an infinite number of optimal vectors lying outside the convex hull, case (ii) above, such as the following perturbation of Example 8:

Example 11. Let us consider rows $r_1 = (0.31, 0.19, 0.2, 0.3)$, $r_2 = (0.2, 0.3, 0.2, 0.3)$, and $r_3 = (0.2, 0.2, 0.3, 0.3)$. We get $[u, v] = ([0.2, 0.31], [0.19, 0.3], [0.2, 0.3], 0.3)$, and $\beta = 0.055$. $R = (0.245, 0.255, [0.245, 0.255], [0.245, 0.355]) = (0.245, 0.255, [0.245, 0.255], [0.245, 0.255])$, as the 4th element of r_1, r_2, r_3 are all the same and greater than all the values in the 4th element of R.

4.4 Summary

In this chapter, we have given a geometric and intuitive insight into both how the virtual abstraction is produced, and how its shape is dictated by the concrete model, and the different forms the shape can take. All three cases above can occur and it would seem to depend strongly on the shape of the convex hull of the concrete rows, which in turn dictates the error value β and the shape of R. We have also given insight into the geometry of the steps involved in converting the concrete model to a virtual MDP, for example the Tight Interval Algorithm, and the loosening of the interval in the case where $[u, v]_{opt}$ is empty.

Chapter 5

Error Propagation

Our method utilises IMDPs induced by a collection of optimal virtual points. We have already seen that by constructing such a virtual IMDP from a DTMC we can reduce the one-step error compared to any lumped DTMC constructed via then naive approach. In practice, the properties we wish to study are not just measurements of the the one-step probabilities of the concrete model, rather, we will be interested in the long term behaviour of the concrete model. Such properties will include those that include a bounded until operator, or more specifically bounded global, or bounded reachability properties. Understanding how the error introduced by abstraction propagates at increasing time steps is key if we are going to attempt to use our abstraction for model checking large DTMCs; Our focus in this chapter is hence on studying how the error of these virtual IMDPs evolve compared to the received APB method, and we will see that for various bounds the decreased one step error bound leads to decreased error bounds across all finite time-frames.

In the first section of this chapter we examine results from [5] that develop bounds on the probabilistic realisation distance induced by an APB, and see whether these can be applied to the new virtual construction, as they were for naive APBs in this same work. Then we produce two bounds on the safety property error of our model; one that is derived from the recursively defined valuation functions which evolves linearly in the associated error of the virtual IMDP, and one that evolves negative-exponentially in the error (derived from [20]). By use of the case study from Chapter 2, we give a comparison of these error bounds against similar ones derived for the safety property error of the model produced by the naive approach. Finally, we develop a more general bound on the propagated error of bounded until formulae, which leverages the Bellman equations for PCTL bounded until. We then again use the case study to compare this bound for the new model against a similarly derived bound for the naive abstraction.

5.1 Probabilistic realisation distance

When we check the properties over the virtual IMDP, we replace the probabilistic operator, $\mathbb{P}$ with the operator $p_{\max}$ or $p_{\min}$, as we have adversarial control over the virtual IMDP and hence can choose the adversary that gives us either of these probabilities. Furthermore, as we need only consider the vertices of the polytope describing the transition probabilities in the IMDP, the adversary giving this maximum (and minimum) probability is calculable. As we know from Theorem 1, there is a deterministic, memoryless adversary that achieves the maximum (minimum) probability of a given PCTL property holding over any possible adversary. What this tells us is that for a given property with an outer probabilistic operator, there is a DTMC induced by the memoryless adversary applied to the IMDP on which the probability of that property holding is the maximum (minimum) probability of that property holding on the virtual IMDP. Furthermore, this DTMC is a APB of the concrete model in question. We can hence attempt to leverage the existing literature, in this case [5], on how error propagates through APBs to study our virtual constructions.

Recall Definitions 16 and 17 of approximate probabilistic bisimulation and probabilistic realisation distance.

Definition 42. Given a DTMC $\mathcal{D} = (Q, q_0, P, L)$ and an APB Γ_ε, we define *the associated APB graph* $\mathcal{G} := (Q, \Gamma_\varepsilon)$. We will ignore loops on vertices given by the reflexivity of Γ_ε, and the symmetry of Γ_ε allows us to consider $\mathcal{G}$ as an undirected graph.

The radius of a connected component of a graph is the minimum eccentricity of any of its vertices. The eccentricity of a vertex is its greatest minimum distance to any other vertex in the connected component.

Definition 43. The *coefficient of ergodicity* of a stochastic matrix T is defined to be:

$$\mathcal{T}(T) := \frac{1}{2}\max_{i,j}\|a_i - a_j\|$$

where a_i, a_j are the i-th and j-th rows of T, and $\|\cdot\|$ is the standard 1-norm.

Further we define the coefficient of ergodicity for a transition set $[\Pi]$ to be $\mathcal{T}([\Pi]) := \underset{T\in[\Pi]}{max}\mathcal{T}(T)$. The coefficient of ergodicity of a DTMC is the coefficient of ergodicity of its transition probability matrix.

Assume that we are given a DTMC $\mathcal{D} = (Q, q_0, P, L)$, and that we are given a partition of the state-space, determined by the labelling, $\mathcal{A} = \{Q_1, \ldots, Q_m\}$, with $q_0 \in Q_1$. By the new method, we produce the virtual IMDP corresponding to $\mathcal{D}$, $\mathcal{I} = (\mathcal{A}, Q_1, [\Pi], L)$. Let us arbitrarily choose a vertex of each of the polytopes corresponding to the rows of $[\Pi]$, and denote these points $v_1, \ldots, v_m$, respectively. Let $\mathcal{V}$ denote the set of these points. We then have transition probability matrix R, with $R_{ij} = v_{ij}$; the probability according to

the virtual point v_i of transitioning from Q_i to Q_j. We have errors of $\beta_1, \ldots, \beta_m$ further corresponding to each of these virtual points. Let $\beta_{max} = \max_{1 \leq i \leq m} \beta_i$.

Let $\tilde{\mathcal{M}} = (Q \cup \mathcal{V}, q_0, \tilde{P}, L)$ be a DTMC, where $\tilde{P} = \left(\begin{smallmatrix} P & 0 \\ 0 & R \end{smallmatrix}\right)$. We can define on this matrix the following APB:

$$\Gamma = \bigcup_{1 \leq i \leq m} \{(s, v_i), (v_i, s), (v_i, v_i), (s, s) \mid s \in Q_i\}.$$

Our construction ensures that this is an APB with error equal to β_{max}. Now we have a lumping for this DTMC, $\tilde{\mathcal{A}} = \{\tilde{Q_1}, \ldots, \tilde{Q_m}\}$, where $\tilde{Q_i} = Q_i \cup \{v_i\}$, which is the set of Γ-closed sets. The associated APB graph consists of m connected components, each with a v_i as its central vertex, which has edges to each $s \in Q_i$, and there are no other edges. Thus the radius of each connected component is 1.

We are thus interested in the following distance for all $1 \leq i, j \leq m$, $s \in Q_i$:

$$|\tilde{P}^k(v_i, \tilde{Q_j}) - \tilde{P}^k(s, \tilde{Q_j})| = |R^k(v_i, v_j) - P^k(s, Q_j)|.$$

We know that $|R(v_i, v_j) - P(s, Q_j)| \leq \beta_{max}$ by the definition of an APB, so for any $(s, s') \in \Gamma$, $d^1_\Gamma(s, s') \leq \beta_{max}$.

Using the following result from [5], we get a conservative bound on the probabilistic realisation distance for arbitrary $k \in \mathbb{N}$:

Theorem 5. Given a DTMC $\mathcal{D} = (Q, q_0, P, L)$ and an APB Γ with error ε, then for any $(s, s') \in \Gamma$ and for any $k \in \mathbb{N}$, the following holds:

$$d^k_\Gamma(s, s') \leq \varepsilon \left(m + \varepsilon \sum_{i=1}^{m} r_i \right)^{k-1},$$

where m is the number of connected components of the associated APB graph, and the r_i are the respective radii of each connected component.

Using this theorem, we can immediately establish that for our model, for any $k \in \mathbb{N}$, for any $(s, s') \in \Gamma$, $d^k_\Gamma(s, s') \leq \beta_{max} \left(m + \beta_{max} \sum_{i=1}^{m} 1\right)^{k-1} = \beta_{max} \left(m + m\beta_{max}\right)^{k-1}$. This bound will increase monotonically over an infinite horizon, and hence in order to try and improve this bound, [5] also provides the following:

Theorem 6. Given a DTMC $\mathcal{D} = (Q, q_0, P, L)$ and an APB Γ with error ε, for any $(s, s') \in \Gamma$ and $k \in \mathbb{N} \cup \{\infty\}$, the following bound holds:

$$d^k_{\Gamma_\varepsilon}(s, s') \leq \tau^k + \varepsilon \lambda m \sum_{i=0}^{k-1} \tau^i$$

where m is the number of connected components of the associated APB graph, λ is the maximum diameter of any connected component of the graph, and $\tau = \mathcal{T}(\mathcal{M}) + \varepsilon \lambda m$.

Now in the case of our model, we have $\varepsilon = \beta_{max}$, $\lambda = 2$. However because of the shape of $\tilde{P} = \left(\begin{smallmatrix} P & 0 \\ 0 & R \end{smallmatrix}\right)$, we have that $\mathcal{T}(\mathcal{D})$ is exactly 1. Hence in our current construction, this bound will still not converge. Furthermore, it will always be greater than 1, which is too conservative to give any meaningful information about the propagation of errors in the new construction. We know that $\mathcal{T}(\tilde{P}) \geq \mathcal{T}(P)$, and it may be possible that we can augment $\tilde{P}$ such that this lower bound is achieved. However the convergence of the expression in the above theorem would still depend on the transition probability matrix of the concrete model being sufficiently ergodic; i.e. with ergodicity $\mathcal{T}(P) < 1 - 2m\beta_{max}$.

5.2 Safety property error

The following material makes use of [21]. Consider a DTMC $\mathcal{D} = (Q, q_0, P, L)$, and consider finite time bounded safety formula $\mathbb{P}^{\mathcal{D}}_{=?}[\mathbf{G}^{\leq N}\phi]$, where $N \in \mathbb{N}$ and ϕ is a boolean function of the labels of the DTMC. (We use the superscript on $\mathbb{P}$ simply to indicate that here we are measuring the probability of this property holding over $\mathcal{D}$.) The specification can be characterised by the following recursive equations, for any $s \in Q$:

$$V_0(s) = \mathbb{1}_\phi(s)\pi_0(s), \quad V_{k+1}(s) = \mathbb{1}_\phi(s) \sum_{s' \in Q} V_k(s')P(s', s)$$

so that $\mathbb{P}^{\mathcal{D}}_{=?}[\mathbf{G}^{\leq N}\phi] = \sum_{s \in Q} V_N(s)$. Here $\mathbb{1}_\phi(s)$ is the indicator function that is 1 if and only if $s \vDash \phi$, 0 otherwise. π_0 is the initial distribution. In our case, as we consider DTMCs to have a given initial state, q_0, we have that $\pi_0(s) = 1$ if and only if $s = q_0$, and equals 0 otherwise. Let us now consider the virtual MDP that we construct via our method that corresponds to $\mathcal{D}$, $\mathcal{M} = (\mathcal{A}, Q_1, \delta, L)$. We have the following recursive functions that characterise the specification $\mathbb{P}^{\mathcal{M}}_{min=?}(\mathbf{G}^{\leq N}\phi)$ over $\mathcal{M}$; for any $Q_i \in \mathcal{A}$:

$$V'_0(Q_i) = \mathbb{1}'_\phi(Q_i)\pi'_0(Q_i), \quad V'_{k+1}(Q_i) = \mathbb{1}'_\phi(Q_i) \sum_{Q_j \in \mathcal{A}} V'_k(Q_j) \min_{R \in \delta(Q_j)} R(Q_j, Q_i)$$

so that $\mathbb{P}^{\mathcal{M}}_{\text{min}=?}[\mathbf{G}^{\leq N}\phi] = \sum_{Q_i \in \mathcal{A}} V'_N(Q_i)$. Here, $R(Q_j, Q_i)$ is the entry of $R \in \delta(A_j)$ corresponding to the probability of moving to the state Q_i. As we are interested in safety, we first focus on finding the minimum probability that our approximation gives of staying safe, hence why we take the minimum of the possible transition probability vectors at each state.

We are interested in the difference between these two probabilities, and so interested in characterising the following distance:

$$\left| \sum_{s \in Q} V_N(s) - \sum_{Q_i \in \mathcal{A}} V'_N(Q_i) \right| = \left| \sum_{Q_i \in \mathcal{A}} \left(V'_N(Q_i) - \sum_{s \in Q_i} V_N(s) \right) \right|.$$

We will refer to this error as the safety property error (at time N), and denote it as μ_N.

Theorem 7. Consider a DTMC $\mathcal{D} = (Q, q_0, P, L)$ and its corresponding virtual MDP $\mathcal{M} = (\mathcal{A}, Q_1, \delta, L)$ with associated error β. Given $\mathbf{G}^{\leq N}\phi$, where $N \in \mathbb{N}$ and ϕ is a boolean function of the labels of the DTMC, the safety property error is bounded as follows:

$$\mu_N = \left| \sum_{s \in Q} V_N(s) - \sum_{Q_i \in \mathcal{A}} V'_N(Q_i) \right| \leq |\mathcal{A}| \beta \sum_{i=0}^{N} \sum_{Q_j \in \mathcal{A}} V'_i(Q_j).$$

Proof. For the base case of the induction, we see that for all $1 \leq i \leq m$, $|V'_0(Q_i) - \sum_{s \in Q_i} V_0(s)| = 0$, as they share labels, and Q_i is the initial state only if it contains q_0. So it follows that $\mu_0 = 0$, which satisfies the bound. Now for the inductive case:

$$\mu_{k+1} = \left| \sum_{Q_i \in \mathcal{A}} \left(V'_{k+1}(Q_i) - \sum_{s \in Q_i} V_{k+1}(s) \right) \right| =$$

$$\left| \sum_{Q_i \in \mathcal{A}} \left(\mathbb{1}'_\phi(Q_i) \sum_{Q_j \in \mathcal{A}} V'_k(Q_j) \min_{R \in \delta(Q_j)} R(Q_j, Q_i) - \sum_{s \in Q_i} \mathbb{1}_\phi(s) \sum_{s' \in Q} V_k(s') P(s', s) \right) \right| =$$

$$\left| \sum_{Q_i \in \mathcal{A}} \mathbb{1}'_\phi(Q_i) \left(\sum_{Q_j \in \mathcal{A}} V'_k(Q_j) \min_{R \in \delta(Q_j)} R(Q_j, Q_i) - \sum_{s \in Q_i} \sum_{s' \in Q} V_k(s') P(s', s) \right) \right| =$$

$$\left| \sum_{Q_i \in \mathcal{A}} \mathbb{1}'_\phi(Q_i) \left(\sum_{Q_j \in \mathcal{A}} V'_k(Q_j) \min_{R \in \delta(Q_j)} R(Q_j, Q_i) - \sum_{s' \in Q} V_k(s') P(s', Q_i) \right) \right| \leq$$

$$\left| \sum_{Q_i \in \mathcal{A}} \left(\sum_{Q_j \in \mathcal{A}} V'_k(Q_j) \min_{R \in \delta(Q_j)} R(Q_j, Q_i) - \sum_{s' \in Q} V_k(s') P(s', Q_i) \right) \right|.$$

Inside the outermost sum, we add and take away the following term:

$$\sum_{Q_j \in \mathcal{A}} V'_k(Q_j) P(s_j, Q_i),$$

where s_j is some arbitrary element of A_j. Hence the above is less than or equal to:

$$\left| \sum_{Q_i \in \mathcal{A}} \left(\sum_{Q_j \in \mathcal{A}} V'_k(Q_j) \min_{R \in \delta(Q_j)} R(Q_j, Q_i) - \sum_{Q_j \in \mathcal{A}} V'_k(Q_j) P(s_j, Q_i) \right) \right| +$$

$$\left| \sum_{Q_i \in \mathcal{A}} \left(\sum_{Q_j \in \mathcal{A}} V'_k(Q_j) P(s_j, Q_i) - \sum_{s' \in Q} V_k(s') P(s', Q_i) \right) \right|.$$

The first expressions is equal to:

$$\left|\sum_{Q_i\in\mathcal{A}}\left(\sum_{Q_j\in\mathcal{A}}V'_k(Q_j)\left(\min_{R\in\delta(Q_j)}R(Q_j,Q_i)-P(s_j,Q_i)\right)\right)\right|\leq$$
$$\left|\sum_{Q_i\in\mathcal{A}}\left(\beta\sum_{Q_j\in\mathcal{A}}V'_k(Q_j)\right)\right|=$$
$$|\mathcal{A}|\beta\sum_{Q_j\in\mathcal{A}}V'_k(Q_j).$$

The second expression is equal to:

$$\left|\sum_{Q_i\in\mathcal{A}}V'_k(Q_i)-\sum_{s\in Q}V_k(s)\right|=\mu_k.$$

Therefore, we get by recursion that:

$$\mu_{k+1}=\left|\sum_{Q_i\in\mathcal{A}}\left(V'_{k+1}(Q_i)-\sum_{s\in Q_i}V_{k+1}(s)\right)\right|\leq$$
$$|\mathcal{A}|\beta\sum_{Q_j\in\mathcal{A}}V'_k(Q_j)+\mu_k\leq$$
$$|\mathcal{A}|\beta\sum_{i=0}^{k}\sum_{Q_j\in\mathcal{A}}V'_i(Q_j).$$

Hence we obtain the bound for the case where $k+1=N$, as required. □

The safety property error grows at most linearly in the error of the virtual MDP as N increases. The structure of this proof is such that we get the following corollaries:

Theorem 8. Consider a DTMC $\mathcal{D}=(Q,q_0,P,L)$ and its corresponding virtual MDP $\mathcal{M}=(\mathcal{A},Q_1,\delta,L)$ with associated error β. Suppose we are given a PCTL formula $\mathbf{G}^{\leq N}\phi$, where $N\in\mathbb{N}$ and ϕ is a boolean function of the labels of the DTMC. Suppose further that we redefine the recursive functions V'_k as follows:

$$V'_0(Q_i)=\mathbb{1}'_\phi(Q_i)\pi'_0(Q_i),\quad V'_{k+1}(Q_i)=\mathbb{1}'_\phi(Q_i)\sum_{Q_j\in\mathcal{A}}V'_k(Q_j)\max_{R\in\delta(Q_j)}R(Q_j,Q_i)$$

Then the (maximum version of the) safety property error is bounded as follows:

$$\mu_N=\left|\sum_{s\in Q}V_N(s)-\sum_{Q_i\in\mathcal{A}}V'_N(Q_i)\right|\leq|\mathcal{A}|\beta\sum_{i=0}^{N}\sum_{Q_j\in\mathcal{A}}V'_i(Q_j).$$

Proof. Simply replace all occurrences of "min" with "max" in the proof of Theorem 7. □

Theorem 9. Consider a DTMC $\mathcal{M} = (Q, q_0, P, L)$ and its corresponding lumped DTMC $\tilde{\mathcal{D}} = (\mathcal{A}, Q_1, \tilde{P}, L)$ obtained via the naive abstraction approach, with associated error β'. Suppose we are given a PCTL formula $\mathbf{G}^{\leq N}\phi$, where $N \in \mathbb{N}$ and ϕ is a boolean function of the labels of the DTMC. Suppose further that we redefine the recursive functions V_k' as follows:

$$V_0'(Q_i) = \mathbb{1}_\phi'(Q_i)\pi_0'(Q_i), \quad V_{k+1}'(Q_i) = \mathbb{1}_\phi'(Q_i) \sum_{Q_j \in \mathcal{A}} V_k'(Q_j)\tilde{P}(Q_j, Q_i)$$

Then the safety property error for the lumped DTMC is bounded as follows:

$$\mu_N = \left| \sum_{s \in Q} V_N(s) - \sum_{Q_i \in \mathcal{A}} V_N'(Q_i) \right| \leq |\mathcal{A}|\beta' \sum_{i=0}^{N} \sum_{Q_j \in \mathcal{A}} V_i'(Q_j).$$

Proof. Replace all occurrences of "$\min\limits_{R \in \delta(Q_j)} R(Q_j, Q_i)$" with "$\tilde{P}(Q_j, Q_i)$" in the proof of Theorem 7. □

Corollary 1. Consider a safety property $\mathbf{G}^{\leq N}\phi$, and the two sets of recursive functions corresponding to the minimum and maximum formulations of the safety property error, which we write as follows:

$$V_0^{\min}(Q_i) = \mathbb{1}_\phi'(Q_i)\pi_0'(Q_i), \quad V_{k+1}^{\min}(Q_i) = \mathbb{1}_\phi'(Q_i) \sum_{Q_j \in \mathcal{A}} V_k^{\min}(Q_j) \min_{R \in \delta(Q_j)} R(Q_j, Q_i)$$

and,

$$V_0^{\max}(Q_i) = \mathbb{1}_\phi'(Q_i)\pi_0'(Q_i), \quad V_{k+1}^{\max}(Q_i) = \mathbb{1}_\phi'(Q_i) \sum_{Q_j \in \mathcal{A}} V_k^{\max}(Q_j) \max_{R \in \delta(Q_j)} R(Q_j, Q_i).$$

Then, for the safety property error bounds given by Theorem 7 and Theorem 8, we have that:

$$|\mathcal{A}|\beta \sum_{i=0}^{N} \sum_{Q_j \in \mathcal{A}} V_i^{\min}(Q_j) \leq |\mathcal{A}|\beta \sum_{i=0}^{N} \sum_{Q_j \in \mathcal{A}} V_i^{\max}(Q_j)$$

Proof. For each $0 \leq i \leq N$:

$$\sum_{Q_j \in \mathcal{A}} V_i^{\min}(Q_j) = \mathbb{P}_{\min=?}[\mathbf{G}^{\leq i}\phi] \leq \mathbb{P}_{\max=?}[\mathbf{G}^{\leq i}\phi] = \sum_{Q_j \in \mathcal{A}} V_i^{\max}(Q_j).$$

Hence the result follows. □

5.3 Negative exponential bounds

In the above section concerning the safety property error, we found we could put a bound on the error that evolved linearly in the value of β, the associated error of the different abstractions. Here we leverage work done in [20] to improve this to a negative exponential function of β. So again consider a DTMC $\mathcal{D} = (Q, q_0, P, L)$, and a finite time bounded safety formula $\mathbb{P}^{\mathcal{D}}_{=?}[\mathbf{G}^{\leq N}\phi]$, where $N \in \mathbb{N}$ and ϕ is a boolean function of the labels of the DTMC. Firstly, for the naive approach, the next result follows from Theorems 2 and 4 in [20]:

Theorem 10. Given a DTMC $\mathcal{D} = (Q, q_0, P, L)$, and a finite time bounded safety formula $\mathbf{G}^{\leq N}\phi$, where $N \in \mathbb{N}$ and ϕ is a boolean function of the labels of the DTMC, and given a lumped DTMC obtained via the naive approach with associated error ε, $\tilde{\mathcal{D}} = (\mathcal{A}, Q_1, \tilde{P}, L)$, then:

$$\mu_N = \left|\mathbb{P}^{\mathcal{D}}_{=?}[\mathbf{G}^{\leq N}\phi] - \mathbb{P}^{\tilde{\mathcal{D}}}_{=?}[\mathbf{G}^{\leq N}\phi]\right| \leq (1 - (1-\varepsilon)^N)$$

This result will also hold for the virtual abstraction that we construct, and hence we have:

Theorem 11. Given a DTMC $\mathcal{D} = (Q, q_0, P, L)$, and a finite time bounded safety formula $\mathbf{G}^{\leq N}\phi$, where $N \in \mathbb{N}$ and ϕ is a boolean function of the labels of the DTMC, and given its corresponding virtual MDP $\mathcal{M} = (\mathcal{A}, Q_1, \delta, L)$ with associated error β, then both:

$$\mu^{\max}_N = \left|\mathbb{P}^{\mathcal{D}}_{=?}[\mathbf{G}^{\leq N}\phi] - \mathbb{P}^{\mathcal{M}}_{\max=?}[\mathbf{G}^{\leq N}\phi]\right| \leq (1 - (1-\beta)^N)$$

and,

$$\mu^{\min}_N = \left|\mathbb{P}^{\mathcal{D}}_{=?}[\mathbf{G}^{\leq N}\phi] - \mathbb{P}^{\mathcal{M}}_{\min=?}[\mathbf{G}^{\leq N}\phi]\right| \leq (1 - (1-\beta)^N).$$

The reason this holds is for reasons that we have discussed before. We know that there are memoryless adversaries for which the probabilities of the property $\mathbf{G}^{\leq N}\phi$ holding are equal to the minimum and maximum probabilities across all possible adversaries ([1]). As these adversaries are memoryless, the DTMC induced by them is finite state, and furthermore in the case of those obtained from our virtual MDP, will be β-error bisimulations of the concrete model. We hence just apply Theorem 10 to these induced DTMCs, and the result follows.

Remark 4. These bounds differ from those given in the previous section in that they are property independent; they give the same bound no matter what property is being compared over the concrete model and its abstraction. We hence expect to see cases where this gives a drastic improvement on the safety property error. Examples of this will be when $\mathbb{P}_{=?}[\mathbf{G}^{\leq N}\phi]$ increases as N increases, or when this value decreases slowly to 0. The negative exponential error bound tends to 1 as N increases, but may do so slower than the linearly growing error bound. On the other hand, we will also see cases where the

negative exponential error bound does not offer an improvement over the linearly evolving error bound. These cases will be those where $\mathbb{P}_{=?}[\mathbf{G}^{\leq N}\phi]$ tends to 0 as N increases and $\sum_{i=0}^{N}\mathbb{P}_{=?}[\mathbf{G}^{\leq N}\phi]$ converges to some $a < 1$ as N increases, or this sum grows much slower than the negative exponential error bound over the time-scale we are interested in. In the following case studies we will see examples where both of these situations occur, but our main focus will be in demonstrating that the error bounds corresponding to our new models are improvements over those obtained using the naive approach.

5.4 Comparing safety property errors: case study

Let us return to the case study we looked at in the previous section. Recall we have an 11 state DTMC as the concrete model, $\mathcal{D} = (Q, q_0, P, L)$, a naive physical abstraction characterised by the lumped matrix:

$$\begin{pmatrix} 0.2 & 0.45 & 0.35 \\ 0.03 & 0.97 & 0 \\ 0.44 & 0.43 & 0.13 \end{pmatrix}$$

and its corresponding virtual MDP with transitions given by the function δ' as follows:

$$\delta'(Q_a) = \{(0.18, 0.48, 0.34), (0.22, 0.44, 0.34)\},$$

$$\delta'(Q_b) = \{(0.01, 0.98, 0.01), (0.02, 0.98, 0)\},$$

$$\delta'(Q_c) = \{(0.44, 0.43, 0.13), (0.45, 0.42, 0.13)\}.$$

Here we will consider two bounded global properties $\mathbf{G}^{\leq k}\neg c$, and $\mathbf{G}^{\leq k}\neg b$, for values of k between 1 and 10. We will first examine how the error bounds given by Theorems 7, 8, and 9 propagate as k increases, in doing so comparing the errors for lumped DTMC, and the values of $p_{\min=?}$ and $p_{\max=?}$ for the virtual MDP. We will then do likewise for the bounds obtained by Theorems 10 and 11, in order to further compare the errors. Figure 5.1 shows the results of querying the probability of $\mathbf{G}^{\leq k}\neg c$ for the different models in PRISM [4]. Note that in this case the values corresponding to the virtual Pmin abstraction in general lie closer to the values on the concrete model than for either the naive abstraction or the virtual Pmax abstraction.

As we can see from Table 5.1, while the bounds on the safety property error are conservative in the cases of the two abstractions, those obtained by using the virtual abstraction, in particular using either the maximum or the minimum versions of the safety property error, are lower than those obtained by using the naive abstraction. This is what we would expect from Theorems 7, 8, and 9. Furthermore, we see that error bound obtained by considering the virtual Pmin is lower than that when considering the virtual Pmax. This is also to be expected from Corollary 1.

Figure 5.1: Probability of $\mathbf{G}^{\leq k}\neg c$ as k increases.

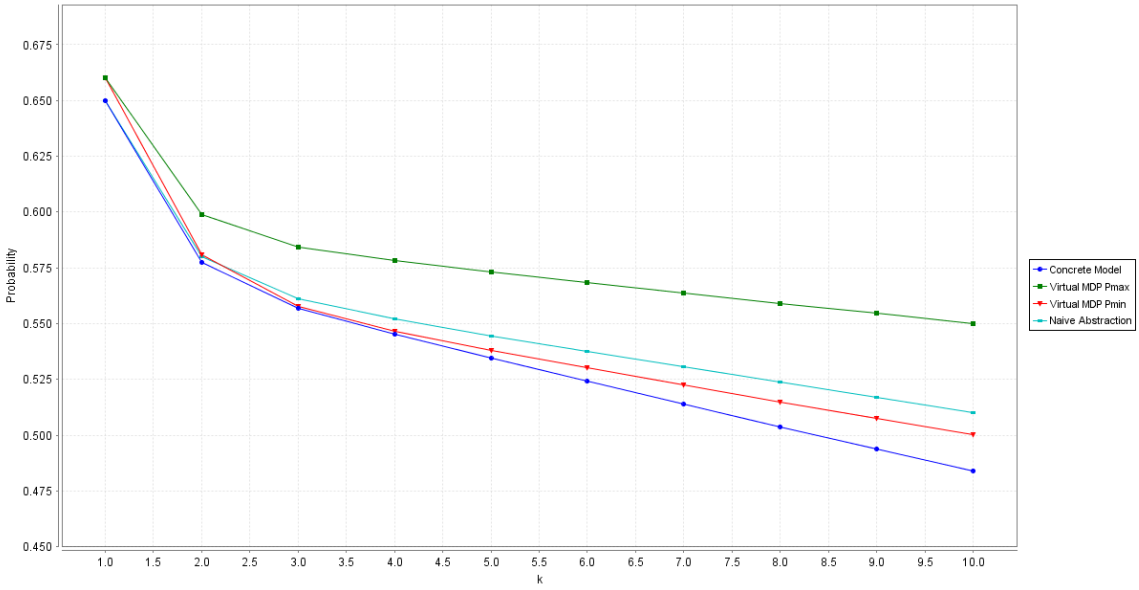
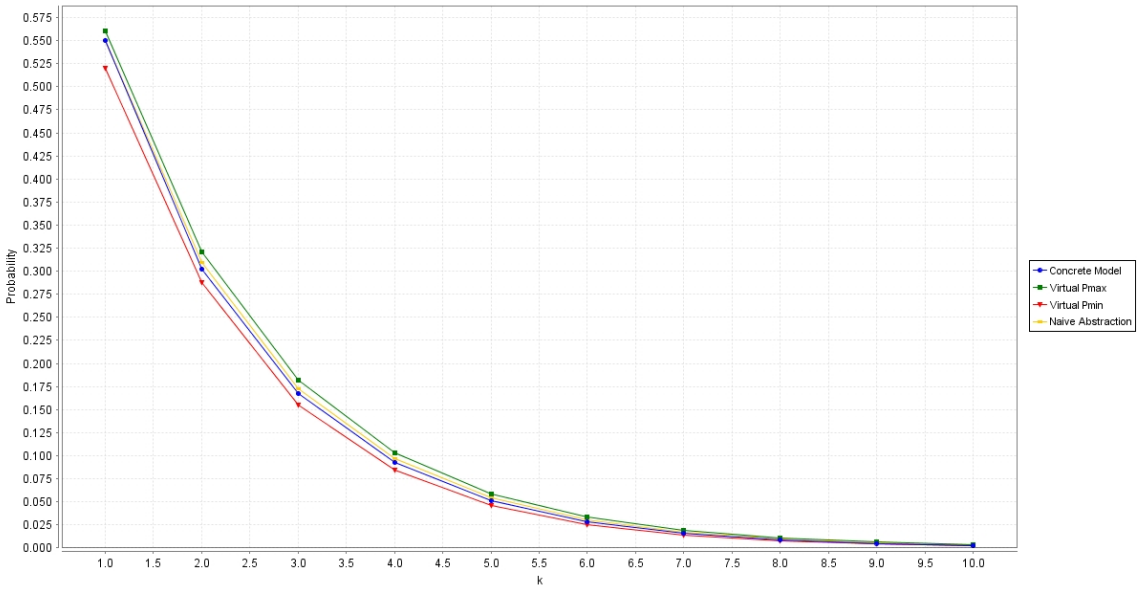


Table 5.1: kth step safety error bound as given by Theorem 7 for $\mathbf{G}^{\leq k}\neg c$.

k	**Virtual Pmax**	**Virtual Pmin**	**Naive Abstraction**
1	0.05	0.05	0.06
2	0.099	0.099	0.117
3	0.18882	0.18612	0.2214
4	0.276498	0.2697552	0.3224295
5	0.363222504	0.351717696	0.421789815
6	0.449198238	0.432408937	0.519805559
7	0.534464443	0.511927947	0.616549442
8	0.619032487	0.590308769	0.712047944
9	0.702909016	0.66757134	0.80631892
10	0.78609984	0.743732396	0.899378483

Figure 5.2: Probability of $\mathbf{G}^{\leq k}\neg b$ as k increases.



In the case of $\mathbf{G}^{\leq k}\neg b$, the results of running PRISM to query the probabilities we are interested in are displayed in Figure 5.2. In this case we see that the naive abstraction happens to be closer to the behaviour of the concrete model.

In Table 5.2 we see that the bounds on the safety property error for the various models evolve similarly as for the case of $\mathbf{G}^{\leq k}\neg c$, with the virtual Pmin consistently giving the best bounds on the safety property error. Also, as the probabilities quickly become close to 0 for this property across all the models, we see that the error bounds are far less conservative that in the case of $\mathbf{G}^{\leq k}\neg c$ as k increases.

We finally examine how the negative-exponential bounds given by Theorems 10 and 11 propagate for these two safety properties, comparing the value for the virtual MDP and the naive abstraction. In this case, as mentioned in Remark 4, we know that the error bounds are property independent, so the errors displayed in Table 5.3 hold for both the properties that we are considering.

Figures 5.3 and 5.4 give a graphical picture of the evolution of all the error bounds we have considered above.

Table 5.2: kth step safety error bounds for $\mathbf{G}^{\leq k}\neg b$.

k	**Virtual Pmax**	**Virtual Pmin**	**Naive Abstraction**
1	0.05	0.05	0.06
2	0.084	0.078	0.099
3	0.13206	0.12111	0.15471
4	0.1593306	0.1443177	0.1857663
5	0.174853974	0.156952569	0.203145759
6	0.183679618	0.16379819	0.212856187
7	0.1886997	0.167514853	0.218285199
8	0.19155464	0.169530936	0.221319708
9	0.193178367	0.170624965	0.223016009
10	0.194101825	0.171218543	0.223964206

Table 5.3: Negative exponential error bounds as k increases.

k	**Virtual MDP**	**Naive Abstraction**
1	0.05	0.06
2	0.0975	0.1164
3	0.142625	0.169416
4	0.18549375	0.21925104
5	0.226219063	0.266095978
6	0.264908109	0.310130219
7	0.301662704	0.351522406
8	0.336579569	0.390431061
9	0.36975059	0.427005198
10	0.401263061	0.461384886

Figure 5.3: Negative exp. and linear saftey error bounds compared on the different models for $\mathbf{G}^{\leq k}\neg c$.

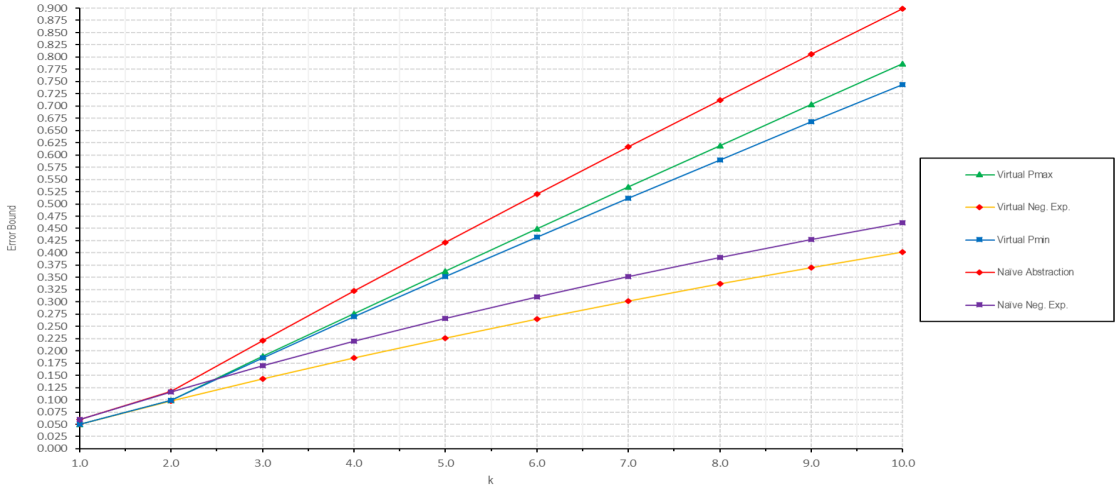
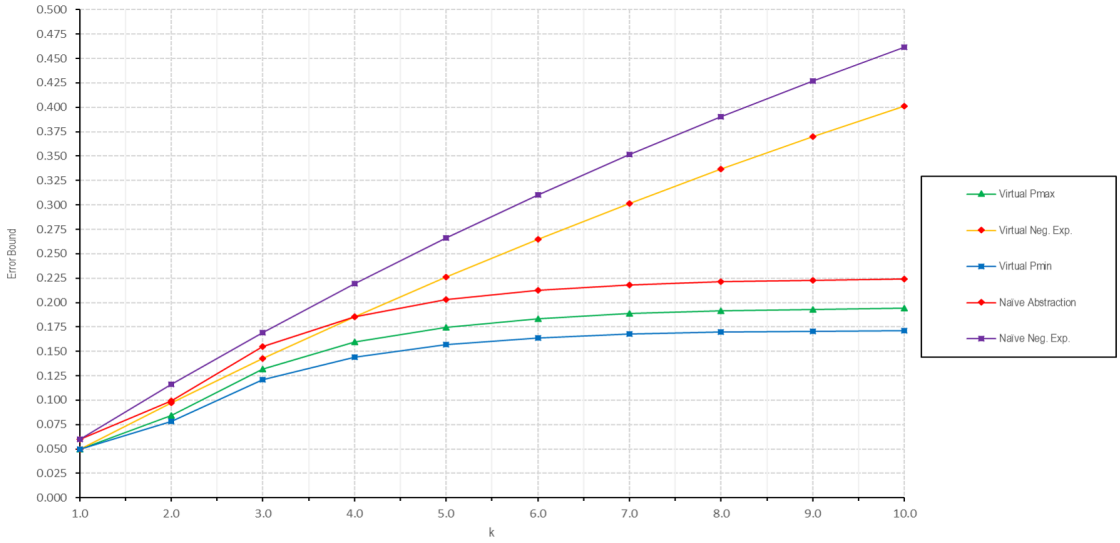


Figure 5.4: Negative exp. and linear saftey error bounds compared on the different models for $\mathbf{G}^{\leq k}\neg b$.



5.5 Error bound on bounded until

For the computation of probabilities of the PCTL bounded until operator for MDPs, we use a set of recursive equations called the Bellman equations ([1], [22]). We consider the case for minimum probabilities. In our models, we give an explicit initial state, but in practice the probabilities for bounded until are computed at all states in parallel, so we consider the probabilities at any state here. Suppose we are given a path formula $\phi \mathbf{U}^{\leq N} \psi$. We are given DTMC $\mathcal{D} = (Q, q_0, P, L)$ and its abstracted virtual MDP $\mathcal{M} = (\mathcal{A}, Q_1, \delta, L)$, with associated error β. For $S \in \{Q, \mathcal{A}\}$, we will denote $S^{yes} = Sat(\psi)$, $S^{no} = S \setminus (Sat(\phi) \cup Sat(\psi)$, and $S^{?} = S \setminus (S^{yes} \cup S^{no})$.

For the DTMC, the probabilities for bounded until are computed using the following equations:

$$\mathbf{Prob}(s, \phi \mathbf{U}^{\leq k} \psi) = \begin{cases} 1 & s \in Q^{yes} \\ 0 & s \in Q^{no} \\ 0 & s \in Q^{?} \text{ and } k = 0 \\ \sum_{s' \in Q} P(s, s') \cdot Prob(s', \phi \mathbf{U}^{\leq k-1} \psi) & \text{otherwise} \end{cases}$$

For the virtual MDP, the probabilities for bounded until are computed using the following equations:

$$p_{\min}(A, \phi \mathbf{U}^{\leq k} \psi) = \begin{cases} 1 & A \in \mathcal{A}^{yes} \\ 0 & A \in \mathcal{A}^{no} \\ 0 & A \in \mathcal{A}^{?} \text{ and } k = 0 \\ \min\limits_{R \in \delta(A)} \sum_{A' \in \mathcal{A}} R(A, A') \cdot p_{\min}(A', \phi \mathbf{U}^{\leq k-1} \psi) & \text{otherwise} \end{cases}$$

We wish to establish a bound on the value of $\left|\mathbf{Prob}(s, \phi \mathbf{U}^{\leq N} \psi) - p_{\min}(A, \phi \mathbf{U}^{\leq N} \psi)\right|$ where $s \in A$, which we will refer to as the Virtual min-Bellman error.

Theorem 12. Given a DTMC $\mathcal{D} = (Q, q_0, P, L)$ and its abstracted virtual MDP $\mathcal{M} = (\mathcal{A}, Q_1, \delta, L)$, with associated error β, and any PCTL bounded until formula $\phi \mathbf{U}^{\leq N} \psi$, for any $s \in Q$, and $A \in \mathcal{A}$ such that $s \in A$, we have the following bound on the Virtual min-Bellman error:

$$\left|\mathbf{Prob}(s, \phi \mathbf{U}^{\leq N} \psi) - p_{\min}(A, \phi \mathbf{U}^{\leq N} \psi)\right| \leq \beta \sum_{j=1}^{N} \sum_{Q_i \in \mathcal{A}} y_{Q_i}^{j-1},$$

where for any $k \in \mathbb{N}$, for any $A \in \mathcal{A}$, we write $y_A^k = p_{\min}(A, \phi \mathbf{U}^{\leq k} \psi)$.

Proof. For any $k \in \mathbb{N}$, any $s \in Q$ we write $x_s^k = \mathbf{Prob}(s, \phi \mathbf{U}^{\leq k} \psi)$, and for any $A \in \mathcal{A}$ we write $y_A^k = p_{\min}(A, \phi \mathbf{U}^{\leq k} \psi)$. We prove the bound holds for any $N \in \mathbb{N}$ by induction. In

the base case, we have that $\left|x_s^0 - y_A^0\right| = 0 - 0 = 0 \leq \beta$. Then, given $k \in \mathbb{N}$, and assuming the bound holds for all $i \leq k$, we get:

$$\left|\mathbf{Prob}(s, \phi \mathbf{U}^{\leq k} \psi) - p_{\min}(A, \phi \mathbf{U}^{\leq k} \psi)\right| = \left|x_s^k - y_A^k\right| =$$

$$\left|\sum_{s' \in Q} P(s, s') \cdot x_{s'}^{k-1} - \min_{R \in \delta(A)} \sum_{Q_i \in \mathcal{A}} R(A, Q_i) \cdot y_{Q_i}^{k-1}\right| =$$

$$\left|\sum_{Q_i \in \mathcal{A}} \sum_{s' \in Q_i} P(s, s') \cdot x_{s'}^{k-1} - \min_{R \in \delta(A)} \sum_{Q_i \in \mathcal{A}} R(A, Q_i) \cdot y_{Q_i}^{k-1}\right|.$$

There is some set $\{s_1, \ldots, s_{|\mathcal{A}|} \mid \forall 1 \leq i \leq |\mathcal{A}|, s_i \in Q_i\}$ such that the above is:

$$\leq \left|\sum_{Q_i \in \mathcal{A}} \sum_{s' \in Q_i} P(s, s') \cdot x_{s_i}^{k-1} - \min_{R \in \delta(A)} \sum_{Q_i \in \mathcal{A}} R(A, Q_i) \cdot y_{Q_i}^{k-1}\right|$$

$$= \left|\sum_{Q_i \in \mathcal{A}} P(s, Q_i) \cdot x_{s_i}^{k-1} - \min_{R \in \delta(A)} \sum_{Q_i \in \mathcal{A}} R(A, Q_i) \cdot y_{Q_i}^{k-1}\right|$$

$$\leq \left|\sum_{Q_i \in \mathcal{A}} P(s, Q_i) \cdot x_{s_i}^{k-1} - \sum_{Q_i \in \mathcal{A}} P(s, Q_i) \cdot y_{Q_i}^{k-1}\right|$$

$$+ \left|\sum_{Q_i \in \mathcal{A}} P(s, Q_i) \cdot y_{Q_i}^{k-1} - \min_{R \in \delta(A)} \sum_{Q_i \in \mathcal{A}} R(A, Q_i) \cdot y_{Q_i}^{k-1}\right|$$

The first expression in this sum is equal to:

$$\left|\sum_{Q_i \in \mathcal{A}} P(s, Q_i) \cdot \left(x_{s_i}^{k-1} - y_{Q_i}^{k-1}\right)\right| \leq$$

$$\max_{Q_i \in \mathcal{A}} \left|x_{s_i}^{k-1} - y_{Q_i}^{k-1}\right|$$

The second expression in this sum is equal to:

$$\left|\sum_{Q_i \in \mathcal{A}} (P(s, Q_i) - R(A, Q_i)) \cdot y_{Q_i}^{k-1}\right| \leq$$

$$\beta \sum_{Q_i \in \mathcal{A}} y_{Q_i}^{k-1}.$$

Hence we get that:

$$\left|x_s^k - y_A^k\right| \leq \beta \sum_{Q_i \in \mathcal{A}} y_{Q_i}^{k-1} + \max_{Q_i \in \mathcal{A}} \left|x_{s_i}^{k-1} - y_{Q_i}^{k-1}\right|$$

and by the inductive hypothesis on k, we get that:

$$\left|x_s^k - y_A^k\right| \leq \beta \sum_{j=1}^{k} \sum_{Q_i \in \mathcal{A}} y_{Q_i}^{j-1}.$$

and thus the bound holds for any $N \in \mathbb{N}$. □

As with the safety property errors, we get the following theorem concerning the Virtual max-Bellman Error, $\left|\mathbf{Prob}(s, \phi \mathbf{U}^{\leq N} \psi) - p_{\max}(A, \phi \mathbf{U}^{\leq N} \psi)\right|$:

Theorem 13. Given a DTMC $\mathcal{D} = (Q, q_0, P, L)$ and its abstracted virtual MDP $\mathcal{M} = (\mathcal{A}, Q_1, \delta, L)$, with associated error β, and any PCTL bounded until formula $\phi \mathbf{U}^{\leq N} \psi$, for any $s \in Q$, and $A \in \mathcal{A}$ such that $s \in A$, we have the following bound on the Virtual max-Bellman error:

$$\left|\mathbf{Prob}(s, \phi \mathbf{U}^{\leq N} \psi) - p_{\max}(A, \phi \mathbf{U}^{\leq N} \psi)\right| \leq \beta \sum_{j=1}^{N} \sum_{Q_i \in \mathcal{A}} y_{Q_i}^{j-1},$$

where for any $k \in \mathbb{N}$, for any $A \in \mathcal{A}$, we write $y_A^k = p_{\max}(A, \phi \mathbf{U}^{\leq k} \psi)$.

Proof. Replace all occurrences of "min" with "max" in the above proof of Theorem 12. □

We also get a likewise bound corresponding to the error for the naive abstraction. We write the lumped naive abstraction as $\tilde{\mathcal{D}} = (\mathcal{A}, Q_1, \tilde{P}, L)$, and consider the Naive Bellman Error, $\left|\mathbf{Prob}^{\mathcal{D}}(s, \phi \mathbf{U}^{\leq N} \psi) - \mathbf{Prob}^{\tilde{\mathcal{D}}}(A, \phi \mathbf{U}^{\leq N} \psi)\right|$:

Theorem 14. Given a DTMC $\mathcal{D} = (Q, q_0, P, L)$ and its corresponding naive abstraction $\mathcal{M} = (\mathcal{A}, Q_1, \tilde{P}, L)$, with associated error β, and any PCTL bounded until formula $\phi \mathbf{U}^{\leq N} \psi$, for any $s \in Q$, and $A \in \mathcal{A}$ such that $s \in A$, we have the following bound on the Naive Bellman error:

$$\left|\mathbf{Prob}^{\mathcal{D}}(q, \phi \mathbf{U}^{\leq N} \psi) - \mathbf{Prob}^{\tilde{\mathcal{D}}}(A, \phi \mathbf{U}^{\leq N} \psi)\right| \leq \beta \sum_{j=1}^{N} \sum_{Q_i \in \mathcal{A}} y_{Q_i}^{j-1},$$

where for any $k \in \mathbb{N}$, for any $A \in \mathcal{A}$, we write $y_A^k = \mathbf{Prob}^{\tilde{\mathcal{D}}}(A, \phi \mathbf{U}^{\leq k} \psi)$.

Proof. Similarly, substitute $\mathbf{Prob}^{\tilde{\mathcal{D}}}$ and $\tilde{P}(\cdot, \cdot)$ for $p_{\min}$ and $R(\cdot, \cdot)$ into the proof of Theorem 12, using the Bellman equations for DTMCs instead of for MDPs. □

Corollary 2. Given a DTMC $\mathcal{D} = (Q, q_0, P, L)$ and its abstracted virtual MDP $\mathcal{M} = (\mathcal{A}, Q_1, \delta, L)$, with associated error β, and any PCTL bounded until formula $\phi \mathbf{U}^{\leq N} \psi$, with for any $k \in \mathbb{N}$, for any $A \in \mathcal{A}$, $y_A^k = p_{\min}(A, \phi \mathbf{U}^{\leq k} \psi)$ we have that:

$$\beta \sum_{j=1}^{N} \sum_{Q_i \in \mathcal{A}} y_{Q_i}^{j-1} \geq N \left| \mathcal{A}^{yes} \right| \beta.$$

Proof. If $Q_i \in \mathcal{A}^{yes}$ then for any $k \in \mathbb{N}$, $y_{Q_i}^k = 1$. Therefore:

$$\beta \sum_{j=1}^{N} \sum_{Q_i \in \mathcal{A}} y_{Q_i}^{j-1} \geq \beta \sum_{j=1}^{N} \sum_{Q_i \in \mathcal{A}^{yes}} y_{Q_i}^{j-1} \geq N \left| \mathcal{A}^{yes} \right| \beta.$$

□

This corollary likewise holds for the Virtual max-Bellman error, and the Naive Bellman error. This tells us that in general these bounds will be conservative, but it is still worth comparing the naive versus the virtual, which we do in the following section.

5.6 Comparing bounded until errors: case study

Lets return to our case study model again, and see how these bounds compare for the bounded until property $(a \vee b)\mathbf{U}^{\leq k} c$ for $k = 1, \ldots, 10$. Once again, the experiment was run in PRISM. The resulting probabilities are displayed in Figure 5.5. Table 5.4 and Figure 5.6 display the Bellman errors corresponding to the different abstractions. As we can see from the graph, and as we would expect from Theorems 12 and 14, both the Virtual min- and Virtual max-Bellman errors grow slower than the Naive Bellman error.

Figure 5.5: Probabilities of $(a \vee b)\mathbf{U}^{\leq k}c$ across the concrete and abstracted models.

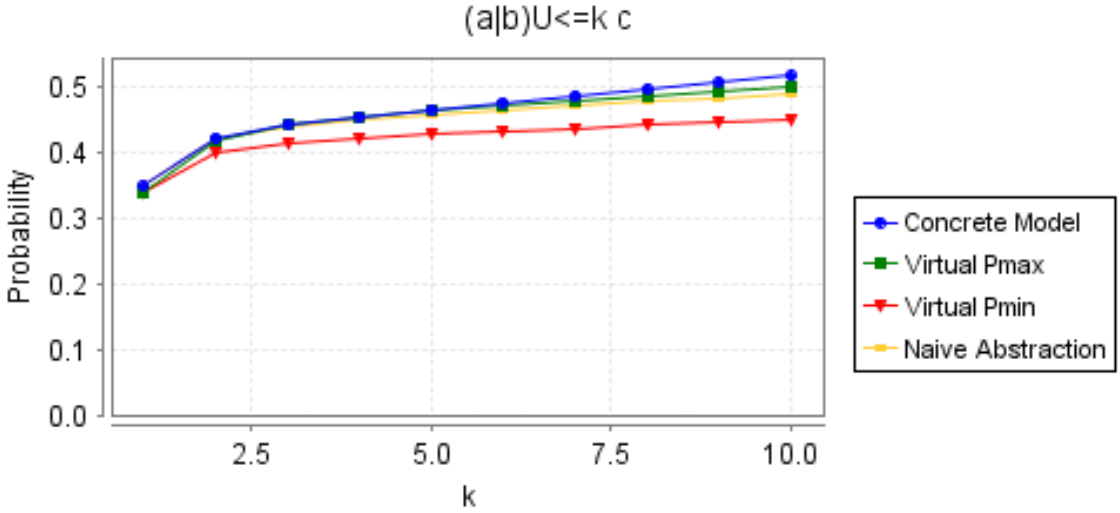
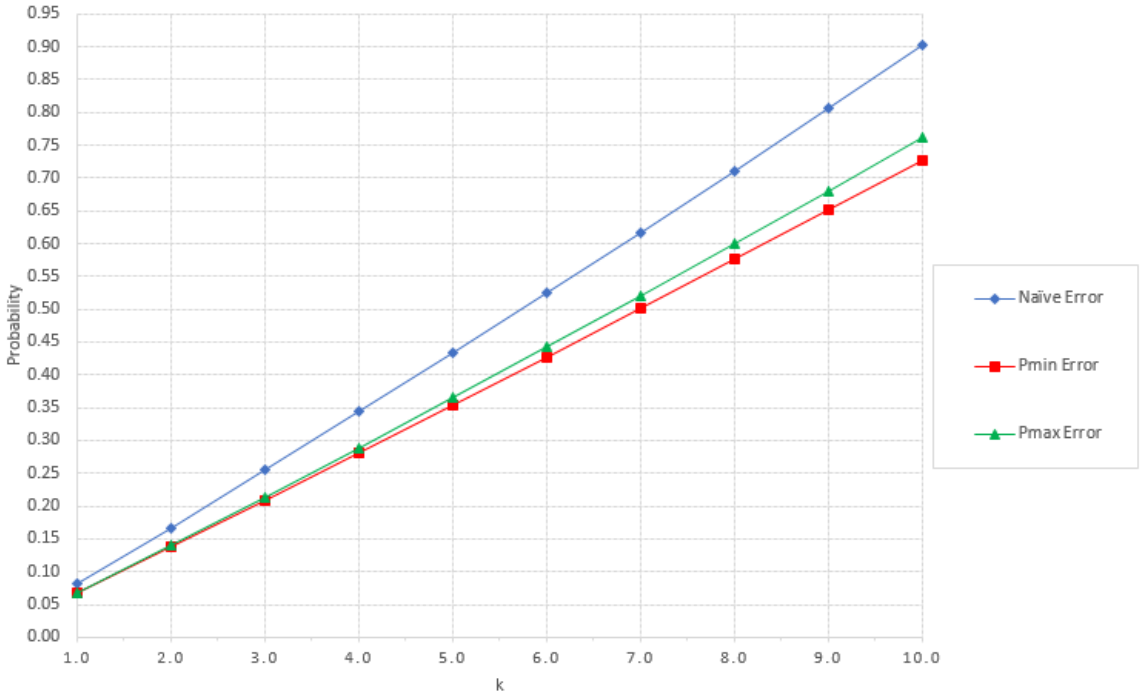


Table 5.4: Bellman errors for $(a \vee b)\mathbf{U}^{\leq k}c$ across the models.

k	**Naive Error**	**Virtual min Error**	**Virtual max Error**
1	0.081	0.067	0.0675
2	0.16683	0.1374	0.13962
3	0.254521	0.208908	0.213588
4	0.343516	0.281135	0.288798
5	0.433703	0.354011	0.365108
6	0.525048	0.42752	0.442475
7	0.617533	0.501654	0.520879
8	0.711142	0.576408	0.600304
9	0.805862	0.651777	0.680736
10	0.901677	0.727757	0.762158

Figure 5.6: Bellman errors for $(a \vee b)\mathbf{U}^{\leq k} c$ across the models.



5.7 Summary

Our early section on the probabilistic realisation distance, following results from [5], gave a brief exposition of these results and showed how they could be applied directly to the new approach. The results that we have produced in the later sections of this chapter have established that for a large fragment of PCTL – that includes all bounded until formulae – the error bounds over finite time frames are smaller for the new approach in comparison to the naive approach, regardless of whether we are considering the value of $p_{\max}$ or $p_{\min}$. In practice, of course, the empirical errors will be smaller still than those estimated conservatively by the bounds here. A point worth making is that in the new abstraction, not all virtual states necessarily do have one-step error equal to the associated error of the abstraction, β. As this value is equal to:

$$\max_{Q_i \in \mathcal{A}} \max_{a,b \in Q_i} |r_a - r_b|_\infty,$$

we see that at some virtual states the error will in actuality be smaller than β.

Conclusion

The main contribution of this work has been to introduce a new method for constructing abstractions of large state space Markov chains with a given partitioning of the state space. In considering sets of optimal virtual points as the representatives of each partition, instead of choosing a "concrete" representative, we have developed an abstraction that both offers improved one-step error in comparison to the received "naive" method of abstraction, and for which the three bounds on the error propagation we introduced are lower than the corresponding bounds for the naive abstraction. Not only does it offer improvements on error bounds, but the computational complexity of model checking by using virtual interval Markov decision processes is prima facie comparable with the complexity of model checking via lumped Markov chains.

Secondly, in Chapter 1 we have produced a thorough literature survey of different interpretations of uncertainty in both Markov chains and Markov decision processes, and seen how these various notions relate to one another. To the extent of our knowledge, no such survey yet exists, and we hope that it may prove a useful contribution to the field of verification.

Further work

There are a number of further questions that arise from the work completed in this project, and a number of points of interest that were beyond our scope. Here, we address directions in which this work could be taken further:

- Determine the exact degree of the polynomial time complexity of model checking via vIMDP. Though we introduced [7] as an improvement on the exponential-time model checking algorithm that uses an equivalent MDP, we did not delve into the details of the polynomial time algorithm from this paper. In practice, if the degree of the polynomial is high, then model checking the vIMDP by this method may still effectively be too computationally inefficient to make our approach a feasible replacement for the received method.

- Produce less conservative bounds on the safety property and Bellman errors for the vIMDP approach. Though the error bounds we produced here are smaller than those

produced for the naive abstraction, they are too conservative to be useful in practice for determining the error of our model after even a small number of time steps.

- Also, we were unable to produce meaningful bounds on the probabilistic realisation distance for our new model. Doing so would allow us, as done in [5], to determine the class of formulae that are robust under the new approach to model checking.
- Implement vIMDP model checking into PRISM or another model checking tool. Doing so would also allow us to make further empirical study into the new approach. Larger concrete models (for example using the PRISM benchmarking suite [23]) could be studied in order to see how the abstraction may perform in practice, as opposed to in our simple case study.

Appendix A

PRISM Specifications

A.1 Properties

```
P=? [ G<=k !'c' ]
Pmax=? [ G<=k !'c' ]
Pmin=? [ G<=k !'c' ]

P=? [ G<=k !'b' ]
Pmax=? [ G<=k !'b' ]
Pmin=? [ G<=k !'b' ]

P=? [ ('a'|'b') U<=k ('c') ]
Pmax=? [ ('a'|'b') U<=k ('c') ]
Pmin=? [ ('a'|'b') U<=k ('c') ]
```

A.2 Models

```
//The concrete model

dtmc

const k;

module concreteModel

    s : [0..10] init 0;
// transition probabilities
    [] s=0 -> 0.05 : (s'=0) + 0.05 : (s'=1) + 0.05 : (s'=2) + 0.05 : (s'=3) +
        0.15 : (s'=4) + 0.15 : (s'=5) + 0.15 : (s'=6) +
        0.30 : (s'=7) + 0.02 : (s'=8) + 0.01 : (s'=9) + 0.02 : (s'=10);

    [] s=1 -> 0.04 : (s'=0) + 0.04 : (s'=1) + 0.05 : (s'=2) + 0.05 : (s'=3) +
        0.14 : (s'=4) + 0.17 : (s'=5) + 0.15 : (s'=6) +
```

```
        0.28 : (s'=7) + 0.03 : (s'=8) + 0.03 : (s'=9) + 0.02 : (s'=10);

    [] s=2 -> 0.01 : (s'=0) + 0.01 : (s'=1) + 0.10 : (s'=2) + 0.05 : (s'=3) +
        0.14 : (s'=4) + 0.15 : (s'=5) + 0.15 : (s'=6) +
        0.20 : (s'=7) + 0.19 : (s'=8);

    [] s=3 -> 0.06 : (s'=0) + 0.04 : (s'=1) + 0.06 : (s'=2) + 0.07 : (s'=3) +
        0.16 : (s'=4) + 0.17 : (s'=5) + 0.15 : (s'=6) +
        0.07 : (s'=7) + 0.03 : (s'=8) + 0.05 : (s'=9) + 0.14 : (s'=10);

    [] s=4 -> 0.01 : (s'=0) + 0.01 : (s'=1) +
        0.96: (s'=4) +
        0.005 : (s'=7) + 0.005 : (s'=8) + 0.005 : (s'=9) + 0.005 : (s'=10);

    [] s=5 -> 0.01 : (s'=1) + 0.01 : (s'=2) + 0.01 : (s'=3) +
        0.01 : (s'=4) + 0.95 : (s'=5) + 0.01 : (s'=6);

    [] s=6 ->0.25 : (s'=4) + 0.50 : (s'=5) + 0.25 : (s'=6);

    [] s=7 -> 0.15 : (s'=0) + 0.15 : (s'=1) + 0.15 : (s'=2) +
        0.15 : (s'=4) + 0.15 : (s'=5) + 0.15 : (s'=6) +
        0.02 : (s'=7) + 0.03 : (s'=8) + 0.03 : (s'=9) + 0.02 : (s'=10);

    [] s=8 -> 0.15 : (s'=0) + 0.15 : (s'=1) + 0.06 : (s'=2) + 0.06 : (s'=3) +
        0.14 : (s'=4) + 0.13 : (s'=5) + 0.15 : (s'=6) +
        0.04 : (s'=7) + 0.03 : (s'=8) + 0.03 : (s'=9) + 0.06 : (s'=10);

    [] s=9 -> 0.40 : (s'=0) + 0.04 : (s'=1) + 0.02 : (s'=2) + 0.01 : (s'=3) +
        0.14 : (s'=4) + 0.13 : (s'=5) + 0.15 : (s'=6) +
        0.10 : (s'=7) + 0.01 : (s'=8);

    [] s=10 -> 0.44 : (s'=0)+
        0.43 : (s'=6) +
        0.13 : (s'=10);

endmodule

//labelling that defines the lumping
label "a" = s=0 | s=1 | s=2 | s=3;
label "b" = s=4 | s=5 | s=6;
label "c" = s=7 | s=8 | s=9 |s=10;
```

```
// the lumped DTMC corresponding to the concrete DTMC.

dtmc

module naiveAbstraction

    Q : [0..2] init 0;
// lumped transition probabilities
    [] Q=0 -> 0.2 : (Q'=0) + 0.45 : (Q'=1) + 0.35 : (Q'=2);
    [] Q=1 -> 0.03 : (Q'=0) + 0.97 : (Q'=1);
    [] Q=2 -> 0.44 : (Q'=0) + 0.43 : (Q'=1) + 0.13 : (Q'=2);

endmodule

// labelling given by the lumping of the concrete model
label "a" = Q=0;
label "b" = Q=1;
label "c" = Q=2;

const k;
```

```
//the vMDP corresponding to the concrete DTMC.

mdp

module vMDP

    Q : [0..2] init 0;
// transition probabilities, with 2 actions at each state
    [] Q=0 -> 0.18 : (Q'=0) + 0.48 : (Q'=1) + 0.34 : (Q'=2);
    [] Q=0 -> 0.22 : (Q'=0) + 0.44 : (Q'=1) + 0.34 : (Q'=2);

    [] Q=1 -> 0.01 : (Q'=0) + 0.98 : (Q'=1) + 0.01 : (Q'=2);
    [] Q=1 -> 0.02 : (Q'=0) + 0.98 : (Q'=1);

    [] Q=2 -> 0.44 : (Q'=0) + 0.43 : (Q'=1) + 0.13 : (Q'=2);
    [] Q=2 -> 0.45 : (Q'=0) + 0.42 : (Q'=1) + 0.13 : (Q'=2);

endmodule

// labelling given by the lumping of the concrete model
label "a" = Q=0;
label "b" = Q=1;
label "c" = Q=2;

const k;
```

References

[1] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.

[2] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, Sep 1994.

[3] C. Baier, B. Haverkort, H. Hermanns, and J. P. Katoen. Model-checking algorithms for continuous-time markov chains. *IEEE Transactions on Software Engineering*, 29(6):524–541, June 2003.

[4] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In G. Gopalakrishnan and S. Qadeer, editors, *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.

[5] A. D'Innocenzo, A. Abate, and J.-P. Katoen. Robust PCTL Model Checking. In *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, Beijing, China, 17-19 April 2012.

[6] S. Derisavi, H. Hermanns, and W. H. Sanders. Optimal state-space lumping in markov chains. *Inf. Process. Lett.*, 87(6):309–315, September 2003.

[7] A. Puggelli, W. Li, A. L. Sangiovanni-Vincentelli, and S. A. Seshia. Polynomial-Time Verification of PCTL Properties of MDPs with Convex Uncertainties. In *Proceedings of the 25th International Conference on Computer Aided Verification*, pages 527–542. Springer-Verlag, 2013.

[8] J. Kemeny, J. Snell, and A. Knapp. *Denumerable Markov Chains*. Springer Verlag, 1976.

[9] D. J. Hartfiel. *Markov Set-Chains*. Springer, 1998.

[10] T. Chen, T. Han, and M Kwiatkowska. On the complexity of model checking interval-valued discrete time Markov chains. *Information Processing Letters*, 113:210–216, 2013.

[11] K. Sen, M. Viswanathan, and G. Agha. Model-Checking Markov Chains in the Presence of Uncertainties. In H. Hermanns and J. Palsberg, editors, *Lecture Notes in Computer Science*, volume 3920, pages 394–410. Springer, 2006.

[12] Y. Zacchia Lun, A. D'Innocenzo, and M. D. Di Benedetto. On stability of time-inhomogeneous Markov jump linear systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 5527–5532, Dec 2016.

[13] E. M. Hahn, H. Hermanns, and L. Zhang. Probabilistic reachability for parametric Markov models. *International Journal on Software Tools for Technology Transfer*, 13(1):3–19, 2011.

[14] R. Givan, S. Leach, and T. Dean. Bounded parameter Markov decision processes. In S. Steel and R. Alami, editors, *Recent Advances in AI Planning: 4th European Conference on Planning, ECP'97 Toulouse, France, September 24–26, 1997 Proceedings*, pages 234–246. Springer Berlin Heidelberg, 1997.

[15] M. Kurano, J. Song, M. Hosaka, and Y. Huang. Controlled Markov set-chains with discounting. *Journal of Applied Probability*, 35(2):293–302, Jun 1998.

[16] C. C. White and H. K. Eldeib. Markov Decision Processes with Imprecise Transition Probabilities. *Operations Research*, 42(4):739–749, 1994.

[17] E. M. Wolff, U. Topcu, and R. M. Murray. Robust control of uncertain markov decision processes with temporal logic specifications. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 3372–3379, Dec 2012.

[18] M. Lahijanian, S. B. Andersson, and C. Belta. Formal verification and synthesis for discrete-time stochastic systems. *IEEE Transactions on Automatic Control*, 60(8):2031–2045, 2015.

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

[20] G. Bian and A. Abate. On the relationship between bisimulation and trace equivalence in an approximate probabilistic context. In *Foundations of Software Science and Computation Structures - 20th International Conference, FOSSACS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*, pages 321–337, 2017.

[21] A. Abate, L. Brim, M. Češka, and M. Kwiatkowska. Adaptive Aggregation of Markov Chains: Quantitative Analysis of Chemical Reaction Networks. In *Computer Aided Formal Verification*. Springer Verlag, 2015.

[22] R. E. Bellman. *Dynamic Programming*. Dover Publications, Inc., 2003.

[23] M. Kwiatkowska, G. Norman, and D. Parker. The PRISM benchmark suite. In *Proc. 9th International Conference on Quantitative Evaluation of SysTems (QEST'12)*, pages 203–204. IEEE CS Press, 2012.