ECAI 2012
Luc De Raedt et al. (Eds.)

doi:10.3233/978-1-61499-098-7-744

Path-Constrained Markov Decision Processes: bridging the gap between probabilistic model-checking and decision-theoretic planning

Florent Teichteil-Königsbuch [1]

Abstract. Markov Decision Processes (MDPs) are a popular model for planning under probabilistic uncertainties. The solution of an MDP is a policy represented as a controlled Markov chain, whose complex properties on execution paths can be automatically validated using stochastic model-checking techniques. In this paper, we propose a new theoretical model, named Path-Constrained Markov Decision Processes: it allows system designers to directly optimize *safe* policies in a single design pass, whose possible executions are guaranteed to satisfy some probabilistic constraints on their paths, expressed in Probabilistic Real Time Computation Tree Logic. We mathematically analyze properties of PC-MDPs and provide an iterative linear programming algorithm for solving them. We also present experiments that illustrate PC-MDPs and highlight their benefits.

1 INTRODUCTION

Computer science is a very large scientific field, which embraces many connected sub-fields that have been widely studied, but often independently. Two of these sub-fields, which both rely on probabilistic discrete-event dynamic systems, have become mature enough to be used in practical, even industrial, applications: probabilistic model-checking [4], and decision-theoretic planning [7]. The former consists in automatically validating some probabilistic linear-time logic formulas for dynamic systems represented as Discrete-Time Markov Chains (DTMCs). For instance, considering the formal dynamic analysis of a nuclear plant during its conception, a designer may want to check if the probability that a given number of reactors fail within a given time interval, is below a given threshold. The latter sub-field is about optimizing the action policy (controller) of an autonomous agent, whose behaviour is assumed to be represented as a set of DTMCs – one for each of its actions –, in order to maximize some long-term reward-based criterion. For instance, if the agent is a software automatically managing the use of reactors, designers may want to find a policy for regulating the power of reactors, which maximizes benefits and clients' demands based on probabilistic long-term demand prediction.

Both sub-fields share some important features. First, they rely on finite-state discrete-time Markov chains: decision-theoretic planning reasons about controllable Markov chains (i.e. one Markov chain per action), whereas probabilistic model-checking is based upon uncontrollable (standard) Markov chains. Second, solving methods rest upon combinatorial search over paths of the underlying Markov chain in order to compute the fixed-point solution of a given update equation, generally using dynamic programming: path-probability update equation for probabilistic model-checking [4], and Bellman equation for decision-theoretic planning [7]. Nevertheless, to the best of our knowledge, both approaches have never been totally unified within a single model and solving framework, compelling designers to incrementally build the system's controller in an optimize-then-validate loop, often using different formalisms.

In this paper, we propose a radically different approach: we provide a theoretical framework and practical algorithm for automatically constructing optimal *and* safe controllers for probabilistic discrete-event controllable systems in a single design pass. More precisely, we search for a policy that maximizes the total discounted rewards gathered by the autonomous agent during its mission, *among* all policies that satisfy a set of given formulas expressed in probabilistic linear-time logic, which actually constrain possible execution paths of the optimized policy. In the nuclear plant example, our approach would allow a system designer to automatically find a reactor controller that maximizes the long-term plant's benefits and clients' demands, while also formally guaranteeing that the probability that a given number of reactors fail within a given time interval by executing this controller, is below a given threshold. We call our new model *Path-Constrained Markov Decision Processes* (PC-MDPs).

In section 2, we present existing models and methods for either optimizing or validating probabilistic discrete-event dynamic systems. In section 3, we propose PC-MDPs, a unified theoretical model for safely optimizing such systems, without needing to validate the optimized policy after the fact. We also discuss related works, some of them considering weaker relations between stochastic model-checking and decision-theoretic planning than we do. In section 4, we mathematically analyze properties of PC-MDPs and provide an iterative linear programming algorithm for solving them. Finally, we conduct some experiments in section 5, and discuss perspectives of our work in section 6.

2 EXISTING APPROACHES TO OPTIMIZING OR VALIDATING PROBABILISTIC DISCRETE-EVENT DYNAMIC SYSTEMS

2.1 Markov Decision Processes

Decision-theoretic planning often assumes that the effects of the actions of a given agent are memoryless, i.e. the probability that the agent goes to a given state when applying a given action only depends on its current state. In practice, this assumption can be removed by adding states to the model, which represent parts of the agent's history. Under the memoryless assumption, a decision-theoretic planning problem can be represented as a Markov Decision Process

[1] Onera — The French Aerospace Lab; F-31055, Toulouse, France; florent.teichteil@onera.fr

(MDP), which is a tuple $\langle S, A, H, T, R\rangle$, such that [7]: S is the finite set of states; A is the finite set of actions; $H \subseteq \mathbb{N}$ is the temporal reasoning horizon; $T : S \times A \times S \to [0;1]$ is the transition function, where for all $(s,a,s') \in S \times A \times S$ and $t \in H$, $T(s,a,s') = Pr(s_{t+1} = s' \mid a_t = a, s_t = s)$; $R : S \times A \times S \to \mathbb{R}$ is the reward function, where for all $(s,a,s') \in S \times A \times S$, $R(s,a,s')$ is the reward gathered by the agent when applying action a in state s and going to state s' at any time step $t \in H$.

Many optimization criteria have been studied for solving MDPs. The total discounted reward criterion is one of the most popular, because it represents a wide class of problems and it also offers practical efficient solving means. It consists in maximizing the value function $V^{\pi}_{\gamma,H} : S \to \mathbb{R}$ over all stationary policies $\pi : S \to A$, defined as: $\forall s \in S, V^{\pi}_{\gamma,H}(s) = E\left[\sum_{t=0}^{H} \gamma^t r_t \middle| \pi, s_0 = s\right]$, where $0 < \gamma < 1$ and r_t is the stochastic reward gathered at time step t by executing π. Special care must be taken when choosing $\gamma = 1$ and $H = +\infty$, because the value function does not need to converge whatever the MDP structure [7, 5].

Some MDP optimization problems, e.g. constrained ones as presented later in this paper, rely on stochastic Markovian policies $\pi : S \times A \to [0;1]$ where, for all states s, actions a and time step t: $\pi(s,a) = Pr(a_t = a | s_t = s)$, i.e. the probability of choosing action a in state s. The transition function of such a policy, noted $T^{\pi} : S \times S \to [0;1]$, is still Markovian: for all states s and s', $T^{\pi}(s,s') = Pr(s_{t+1} = s' | s_t = s, \pi) = \sum_{a \in A} \pi(s,a) T(s,a,s')$. We note Π the set of all stochastic Markovian policies.

Finally, it is sometimes interesting to evaluate the value of a set of initial states weighted by some probabilities. Noting $\alpha : S \to [0;1]$ the initial probability distribution over states, i.e. $\alpha(s) = Pr(s_0 = s), \forall\, s \in S$, the value of α for a given policy π is simply: $V^{\pi}_{\gamma,H}(\alpha) = \sum_{s \in S} \alpha(s) V\pi_{\gamma,H}(s)$.

It is worth noting that the possible stochastic executions of any stationary policy π (optimal or not, deterministic or stochastic) of an MDP, is a Markov chain, whose behavior and properties can be automatically validated using stochastic model-checking techniques.

2.2 Stochastic Model-Checking

Complex properties on paths of a Markov chain can be validated with probabilistic linear-time logics such as *Probabilistic Real Time Computation Tree Logic* (PCTL) [4]. This formalism allows designers to define tree-like path formulas that are resting on logic connectives ($\neg$, $\wedge$, $\vee$, $\to$), boolean state formulas (set of functions $f : S \to \{0,1\}$), and probabilistic *strong until* temporal operators $\mathcal{U}^{\leqslant H}_{\diamond p} : \{0,1\}^S \times \{0,1\}^S \to \{0,1\}^S$, where $\diamond$ is one of $<$, $\leqslant$, $\geqslant$, $>$. As we consider controlled Markov chains in this paper, we will note $\left[\mathcal{U}^{\leqslant H}_{\diamond p}\right]_{\pi}$ such an operator to indicate that it is defined for the Markov chain induced by policy π. The semantics of these operators is as follows: for a given policy π, state s and boolean state formulas f and g, $\left(f \left[\mathcal{U}^{\leqslant H}_{\diamond p}\right]_{\pi} g\right)(s)$ means that there is at least (if $\diamond \in \{\geqslant, >\}$) or at most (if $\diamond \in \{<, \leqslant\}$) a probability p that both g will become true within H time units in some state reachable from s by executing π, and that f will be true from now on in s until g becomes true in some state reachable from s by executing π. This semantics is perhaps easier to understand by reasoning about paths starting in s and executing π. Let Φ^{π}_s be the set of such paths. For a given path $\phi \in \Phi^{\pi}_s$, we note: $\phi(i)$ the i^{th} state of the path, and ϕ_i the sub-path of ϕ starting in $\phi(i)$. Then, $\left(f \left[\mathcal{U}^{\leqslant H}_{\diamond p}\right]_{\pi} g\right)(s)$ is true iff:

$$Pr\left(\exists\, \phi \in \Phi^{\pi}_s, \exists\, 0 \leqslant i \leqslant H : g(\phi(i)) = 1, \forall\, 0 \leqslant j < i, f(\phi(j)) = 1\right) \diamond p \quad (1)$$

If $H = +\infty$, state index i in the previous equation must be strictly less ($<$) than H, meaning that *g must* become true in *finite* time.

As an example, consider the nuclear plant problem mentioned in the introduction. Imagine that we have n reactors running together, whose status is either `ok` or `fail` for each. We are initially in a state s_0 where the status of all reactors is `ok`. For some reasons, designers want that the probability that a given reactor k fails within H time units while reactors i and j are `ok` is less than 10^{-9}, using a given reactor regulation policy π. In PCTL, they will express this property with the formula $\left(f \left[\mathcal{U}^{H}_{\leqslant 10^{-9}}\right]_{\pi} g\right)(s_0)$, where, for all states $s \in S$, $f(s)$ is true iff status of reactors i and j is `ok` in s, and $g(s)$ is true iff status of reactor k is `fail` in s.

Efficient algorithms for evaluating strong until temporal operators have been proposed [6, 8]. Most of them rely on the exact computation of the left-hand side of eq. 1, which they compare with the right-hand side p, in order to know if the strong until operator is true or false from s. By noting $[P^g_f]^{\pi}_H(s)$ this left-hand side, the following dynamic programming equation can be used to compute $[P^g_f]^{\pi}_H(s)$:

$$[P^g_f]^{\pi}_H(s) = \begin{cases} 1 & \text{if } g(s) = 1 \\ 0 & \text{if } (g(s) = 0) \wedge (f(s) = 0 \vee H = 0) \\ \sum_{s' \in S} T^{\pi}(s,s') [P^g_f]^{\pi}_{H-1}(s') & \text{otherwise} \end{cases} \quad (2)$$

Like for the value function in Markov Decision Processes, we can evaluate the probability of a path formula for a given initial probability distribution α over states, as: $[P^g_f]^{\pi}_H(\alpha) = \sum_{s \in S} \alpha(s) [P^g_f]^{\pi}_H(s)$. The corresponding path-formula $f \left[\mathcal{U}^{\leqslant H}_{\diamond p}\right]_{\pi} g$ is also evaluated over α, but no more over individual states. Yet, note that this little extension to standard stochastic model-checking formalisms includes standard definitions, since any single initial state is a particular deterministic initial distribution.

To ease reading, we will equivalently note the following PCTL constraints: $\left(f \left[\mathcal{U}^{\leqslant H}_{\diamond p}\right]_{\pi} g\right)(\alpha) = 1 \Leftrightarrow [P^g_f]^{\pi}_H(\alpha) \diamond p$.

3 A NEW UNIFIED APPROACH: PATH-CONSTRAINED MARKOV DECISION PROCESSES

If a designer wants to optimize an MDP policy under the constraint that some PCTL formulas are satisfied, there is up to now no other solution than optimizing first the policy, then model-checking whether the given PCTL formulas are satisfied, if not "magically" modifying the MDP and re-optimizing again the policy for the new MDP, and so on until the PCTL formulas are all satisfied. Yet, there is no easy "magical" way to modify the MDP in such a way that some previously unsatisfied PCTL formulas become satisfied in the new optimization pass. It seems to us that a new decision-theoretic model, defined as a general constraint optimization problem, is needed to properly optimize total stochastic rewards under constraints on the paths of the controlled Markov chain.

3.1 A new constraint optimization problem

Formally, we define a Path-Constrained MDP (PC-MDP) as a tuple $\langle S, A, T, R, n, \mathcal{H}, \mathcal{F}, \mathcal{G}, \mathcal{P}, \alpha\rangle$, where S, A, T, and R are defined as in standard MDPs (see section 2), and: $n \in \mathbb{N}^*$ is a number of PCTL constraints on some path-formulas, $\mathcal{H} = \{H, H_1, \cdots H_n\}$ is a set of temporal horizons, $\mathcal{F} = \{f_1, \cdots, f_n\}$ and $\mathcal{G} = \{g_1, \cdots, g_n\}$ are sets of boolean state formulas used in some PCTL constraints, $\mathcal{P} = \{p_1, \cdots, p_n\}$ is a set of probabilities on these PCTL constraints, and α is an initial probability distribution over states. We search for a stochastic Markovian policy π^* solution of the following path-constraint optimization problem, named **PCMDP-COP$_{\boldsymbol{\gamma}}$**:

$$\pi^*(s) \in \operatorname{argmax}_{\pi\in\Pi} V^{\pi}_{\gamma,H}(\alpha)$$
$$\text{subject to: } \begin{cases} \left(f_1\left[\mathcal{U}^{H_1}_{\diamond_1 p_1}\right]_{\pi^*} g_1\right)(\alpha) = 1 \Leftrightarrow [P^{g_1}_{f_1}]^{\pi^*}_{H_1}(\alpha) \diamond_1 p_1 \\ \vdots \\ \left(f_n\left[\mathcal{U}^{H_n}_{\diamond_n p_n}\right]_{\pi^*} g_n\right)(\alpha) = 1 \Leftrightarrow [P^{g_n}_{f_n}]^{\pi^*}_{H_n}(\alpha) \diamond_n p_n \end{cases}$$

In this first paper about Path-Constrained MDPs, we will only study the infinite horizon case, where $H = H_1 = \cdots = H_n = +\infty$. From now on, we will note $V^{\pi}_{\gamma,\infty}$ simply as V^{π}_{γ}.

3.2 Relations to other constrained MDP models

Some constrained optimization problems based on MDPs have been studied in the literature. In [1], authors propose to find a policy that optimizes the value function of an MDP, subject to constraints on many other value functions based on different reward structures. Contrary to us, the objective function and the constraints have the same mathematical structure, and can be solved with a single linear program. Moreover, their constraint optimization problem is not related at all to stochastic model-checking of some probabilistic linear-time logic formulas.

From an application viewpoint, works by [2, 3] are closer to ours. In [2], authors consider an MDP viewed as a probabilistic controlled automaton, for which they consider a set of properties that each gives rise to some history-dependent reward if it is satisfied. Based on this model, the authors search for a policy that maximizes the history-dependent rewards associated to these properties. To simplify, this approach can be viewed as a standard unconstrained MDP, whose reward structure is related to multiple path-formulas. In [3], authors study how to find a MDP policy that maximizes a probability vector, where each row corresponds to a given path-formula ; in particular, they produce a policy such that all given linear-time logic formulas are satisfied with sufficient probability. Thus, they tackle an unconstrained multi-objective optimization problem, whose reward structure is again related to some linear-time logic formulas.

Our model is different from all these approaches: we propose to optimize a given value function subject to a set of linear-time logic formulas, but the reward structure associated to the optimized value function is totally independent from the set of these formulas. From a technical point of view, our constraint optimization problem is relatively close to [1], except that our objective function and our constraints have different mathematical structures. In the next section, we propose a solving method inspired by [1], but the different nature of the objective function and the constraints in our problem compel us – among others – to solve a convergent sequel of linear programs (but not a single one as in [1]).

Finally, it is worth noting that the wide class of undiscounted MDPs named GSSPs, recently proposed by [5], is itself a subclass of path-constrained MDPs with $\gamma = 1$: GSSP $\subset$ PCMDP-COP$_1$. Indeed, GSSPs bring back many undiscounted MDP problems to optimizing goal-oriented MDPs under the constraint that the optimized policy must reach the goal with probability 1. This particular constraint can be expressed in PCTL as: $\left(\texttt{true}\,[\mathcal{U}^{+\infty}_{\geqslant 1}]_{\pi^*}\, g\right)(s_0) = 1$, where s_0 is the initial state and g is true only in the goal states.

4 SOLVING PATH-CONSTRAINED MDPs

In order to solve the constraint optimization problem stated in section 3, we need to bring out a numerical quantity that links the objective function and the constraints, so that the value of the objective function is explicitly subject to the constraints. In [1], authors propose the so-called *occupation measure* as such a linking numerical quantity, which allows them to reformulate their constraint optimization problem as a linear program, whose vector of variables is this occupation measure. Intuitively, we can express the accumulated rewards and path probabilities in terms of the probabilistic presence of the agent in a given state for a given randomized policy, which is somehow what the occupation measure represents. We will also build our optimization algorithm upon occupation measures, but differently.

4.1 Occupation measure

Consider a given initial probability distribution α on states, and a given stochastic Markovian policy π. For all state $s \in S$ and action $a \in A$, the *occupation measure* $\mu : S \times A \to [0;1]$ is defined as: $\mu^{\pi}_{\alpha}(s,a) = (1-\gamma)\sum_{t=0}^{+\infty} \gamma^t Pr(s_t = s, a_t = a|\alpha,\pi)$. This series is well-defined because its terms are uniformly bounded by γ^t whose series converge to $1/(1-\gamma)$. This occupation measure is in fact a probability measure, since $\sum_{s\in S, a\in A} \mu^{\pi}_{\alpha}(s,a) = 1$. Policy π can be easily obtained from the corresponding occupation measure using the following equation (see [1]), for all states s and actions a:

$$\pi(s,a) = \begin{cases} \frac{\mu^{\pi}_{\alpha}(s,a)}{\sum_{a'\in A} \mu^{\pi}_{\alpha}(s,a')} & \text{if } \sum_{a'\in A} \mu^{\pi}_{\alpha}(s,a') \neq 0 \\ \textit{any probability} & \text{otherwise} \end{cases} \quad (3)$$

It has been proved in [1], that the occupation measure satisfies the following $|S|(|A|+1)$ constraints:

$$\begin{cases} \sum_{\substack{s'\in S \\ a\in A}} \mu^{\pi}_{\alpha}(s',a)(\delta_s(s') - \gamma T(s',a,s)) = (1-\gamma)\alpha(s), \forall\, s \in S \\ \mu^{\pi}_{\alpha}(s,a) \geqslant 0, \forall\, s \in S, a \in A \end{cases} \quad (4)$$

Now, [1] expresses value function $V^{\pi}_{\gamma}(\alpha)$ in terms of μ^{π}_{α}, using the definition of MDP discounted value functions:

$$V^{\pi}_{\gamma}(\alpha) = \frac{1}{1-\gamma} \sum_{s\in S, a\in A} \mu^{\pi}_{\alpha}(s,a) \sum_{s'\in S} T(s,a,s')R(s,a,s') \quad (5)$$

Obtaining a similar expression for PCTL formula probabilities is not easy for at least two reasons: these probabilities are not defined in terms of averages over some random variables, but in terms of complex properties over paths in the controlled Markov chain (see eq. 1 and 2); they do not depend on any discount factor, yet a discount factor is required to properly define occupation measures, which do not need to be properly defined in the general case for $\gamma = 1$.

4.2 Expressing PCTL probabilities as limits of γ-discounted functions of occupation measures

Consider a given constraint $[P^{g_i}_{f_i}]^{\pi}_{\infty}(\alpha) \diamond_i p_i$ for some $1 \leqslant i \leqslant n$. Let $F_i = \{s \in S : f_i(s) = 1\}$ and $G_i = \{s \in S : g_i(s) = 1\}$. Then, eq. 2 gives rise to the following equation, for all $s \in F_i \cap \overline{G_i}$:

$$[P^{g_i}_{f_i}]^{\pi}_{\infty}(s) = \sum_{s'\in F_i\cap\overline{G_i}} T^{\pi}(s,s')[P^{g_i}_{f_i}]^{\pi}_{\infty}(s') + \tilde{R}_i(s) \quad (6)$$

with $\tilde{R}_i(s) = \sum_{s'\in G_i} T^{\pi}(s,s')$, which can be considered as the reward function of a *path-equivalent MDP* related to the i^{th} constraint; in other terms, a reward is gathered only when g_i becomes true for the first time for all paths where f_i was true before.

Let $\mathcal{X}^{\pi}_i \subseteq F_i \cap \overline{G_i}$ be the set of states from which it exists a path with a positive probability to a state in G_i when executing π: $\mathcal{X}^{\pi}_i = \{s \in F_i \cap \overline{G_i} : [P^{g_i}_{f_i}]^{\pi}_{\infty}(s) > 0\}$. We will not need to explicitly compute $\mathcal{X}^{\pi}_i$ in our approach, but note that simple graph reachability algorithms would be sufficient to obtain it without computing path-probabilities. Obviously, in eq. 6, s' can be summed only over $\mathcal{X}^{\pi}_i$. The following lemma will allow us to get an algebraic expression of PCTL formula probabilities from eq. 6.

Lemma 1. *Let $T^{\pi}_{|\mathcal{X}_i^{\pi}}$ be the sub-matrix of T^{π} defined over $\mathcal{X}_i^{\pi} \times \mathcal{X}_i^{\pi}$. Then, $I_{|\mathcal{X}_i^{\pi}} - T^{\pi}_{|\mathcal{X}_i^{\pi}}$ is invertible and equal to $\sum_{t=0}^{+\infty} \left(T^{\pi}_{|\mathcal{X}_i^{\pi}}\right)^t$.*

Proof. $\mathcal{X}_i^{\pi}$ is a set of states in $F_i \cap \overline{G_i}$ for which there exists a path to states in G_i, so that it constitutes a transient class of the sub-Markov chain defined over $F_i \cup G_i$. Thus, according to classical Markov chain results (see Appendix A of [7]), $I_{|\mathcal{X}_i^{\pi}} - T^{\pi}_{|\mathcal{X}_i^{\pi}}$ is invertible. □

Thanks to this lemma, and noting $W_{|\mathcal{X}_i^{\pi}}$ the sub-vector of a given vector W over $\mathcal{X}_i^{\pi}$, we can find an algebraic expression of $[P_{f_i}^{g_i}]^{\pi}_{\infty|\mathcal{X}_i^{\pi}}$ from eq. 6:

$$[P_{f_i}^{g_i}]^{\pi}_{\infty|\mathcal{X}_i^{\pi}} = \left(I_{|\mathcal{X}_i^{\pi}} - T^{\pi}_{|\mathcal{X}_i^{\pi}}\right)^{-1} \tilde{R}_{i|\mathcal{X}_i^{\pi}} = \sum_{t=0}^{+\infty} \left(T^{\pi}_{|\mathcal{X}_i^{\pi}}\right)^t \tilde{R}_{i|\mathcal{X}_i^{\pi}} \quad (7)$$

However, this algebraic expression is not sufficient for our needs, because: (i) we do not want our constraints to depend on some policy-dependent subset of states ($\mathcal{X}_i^{\pi}$); (ii) the series in the previous equation is convergent but not uniformly, so that we will not be able to express it in terms of occupation measures as done in [1] for the discounted value function. Yet, this algebraic expression happens to be very useful for the following proposition, which nevertheless requires that all F_i sets constitute transient classes of the MDP: i.e. for all executions of all policies, there is no path from $\overline{F_i}$ to F_i. In practice, this restriction is not very annoying, since safety or operability PCTL constraints usually aim at checking that the system will eventually lead to (operability) or never reach (safety) some G_i sets *whatever the intermediate states visited*, meaning that $F_i = S$ most of the time.

Proposition 1. *Assume that, for all $1 \leqslant i \leqslant n$, either $F_i = S$ or F_i is a transient class of the MDP. Let $P_i^{\pi}(\gamma)$ be the vector defined over $F_i \cap \overline{G_i}$ by: $P_i^{\pi}(\gamma) = \sum_{t=0}^{+\infty} \gamma^t \left(T^{\pi}_{|F_i \cap \overline{G_i}}\right)^t \tilde{R}_{i|F_i \cap \overline{G_i}}$, $0 < \gamma < 1$. Then: $\lim_{\gamma \to 1} P_i^{\pi}(\gamma) = [P_{f_i}^{g_i}]^{\pi}_{\infty|F_i \cap \overline{G_i}}$*

Proof. As $F_i = S$ or F_i is a transient class of the MDP, $\left(T^{\pi}_{|F_i \cap \overline{G_i}}\right)^t (s,s') = Pr(s_t = s' | s_0 = s, \pi)$ for all states s and s' in $F_i \cap \overline{G_i}$. Thus, for all states $s \in F_i \cap \overline{G_i}$, we have:

$$\begin{aligned}
P_i^{\pi}(\gamma)(s) &= \sum_{t=0}^{+\infty} \sum_{s' \in F_i \cap \overline{G_i}} \gamma^t Pr(s_t = s' | s_0 = s, \pi) \tilde{R}_i(s') \\
&= \sum_{t=0}^{+\infty} \sum_{s' \in \mathcal{X}_i^{\pi}} \gamma^t Pr(s_t = s' | s_0 = s, \pi) \tilde{R}_i(s') + \\
&\qquad \sum_{t=0}^{+\infty} \sum_{s' \in \overline{\mathcal{X}_i^{\pi}} \cap F_i \cap \overline{G_i}} \gamma^t \underbrace{Pr(s_t = s' | s_0 = s, \pi) \tilde{R}_i(s')}_{A(s',t)=0} \\
&= \sum_{t=0}^{+\infty} \sum_{s' \in \mathcal{X}_i^{\pi}} \gamma^t Pr(s_t = s' | s_0 = s, \pi) \tilde{R}_i(s')
\end{aligned}$$

Indeed, $A(s',t) = 0$ for all $s' \in \overline{\mathcal{X}_i^{\pi}} \cap F_i \cap \overline{G_i}$ and time step t, because by definition of $\mathcal{X}_i^{\pi}$, there is no path from s' to some state in G_i so that $\tilde{R}_i(s') = \sum_{s'' \in G_i} T^{\pi}(s', s'') = 0$. Now, we have to consider two cases, depending on whether s is in $\mathcal{X}_i^{\pi}$ or not. If $s \in \mathcal{X}_i^{\pi}$, we have:

$$P_i^{\pi}(\gamma)(s) = \sum_{t=0}^{+\infty} \gamma^t \left(\left(T^{\pi}_{|\mathcal{X}_i^{\pi}}\right)^t \tilde{R}_{i|\mathcal{X}_i^{\pi}} \right)(s) \xrightarrow[\gamma \to 1]{} [P_{f_i}^{g_i}]^{\pi}_{\infty|\mathcal{X}_i^{\pi}}(s)$$

because the above series is uniformly convergent and thus a continuous function of γ, and using eq. 7. If $s \in \overline{\mathcal{X}_i^{\pi}} \cap F_i \cap \overline{G_i}$, there is no path from s to any state in $\mathcal{X}_i^{\pi}$ (otherwise, there would be a path from s to G_i via some state in $\mathcal{X}_i^{\pi}$, which contradicts the fact that $s \in \overline{\mathcal{X}_i^{\pi}}$): thus, $Pr(s_t = s' | s_0 = s, \pi) = 0$ for all time steps t and states $s' \in \mathcal{X}_i^{\pi}$, so that $P_i^{\pi}(\gamma)(s) = 0 = [P_{f_i}^{g_i}]^{\pi}_{\infty}(s)$; the last equality comes from the definition of $\mathcal{X}_i^{\pi}$. □

Finally, we can express $P_i^{\pi}(\gamma)$ in terms of occupation measures, and thus get a discounted function of occupation measures that converges to PCTL formula probabilities, thanks to the next theorem.

Theorem 1. *Assume that, for all $1 \leqslant i \leqslant n$, either $F_i = S$ or F_i is a transient class of the MDP. Let α be an initial probability distribution over states such that $\alpha(s) = 0$ for all $s \notin F_i \cap \overline{G_i}$. Let be $P^{\pi}_{i,\alpha}(\gamma) = \frac{1}{1-\gamma} \sum_{s \in F_i \cap \overline{G_i}, a \in A} \mu_{\alpha}^{\pi}(s,a) \sum_{s' \in G_i} T(s,a,s')$. Then: $\lim_{\gamma \to 1} P^{\pi}_{i,\alpha}(\gamma) = [P_{f_i}^{g_i}]^{\pi}_{\infty}(\alpha)$*

Proof. For all states $s' \in F_i \cap \overline{G_i}$, we have:

$$\begin{aligned}
\tilde{R}_i(s') &= \sum_{s'' \in G_i} T^{\pi}(s', s'') = \sum_{s'' \in G_i} \sum_{a \in A} \pi(s', a) T(s', a, s'') \\
&= \sum_{s'' \in G_i} \sum_{a \in A} Pr(a_t = a | s_t = s', \pi) T(s', a, s'')
\end{aligned}$$

As π is a Markovian policy: $Pr(s_t = s' | s_0 = s, \pi) Pr(a_t = a | s_t = s', \pi) = Pr(s_t = s', a_t = a | s_0 = s, \pi)$. This yields, $\forall s \in F_i \cap \overline{G_i}$:

$$\begin{aligned}
P_i^{\pi}(\gamma)(s) &= \sum_{t=0}^{+\infty} \sum_{s' \in F_i \cap \overline{G_i}} \gamma^t Pr(s_t = s' | s_0 = s, \pi) \tilde{R}_i(s') \\
&= \sum_{\substack{s' \in F_i \cap \overline{G_i} \\ a \in A}} \sum_{t=0}^{+\infty} \gamma^t Pr(s_t = s', a_t = a | s_0 = s, \pi) \sum_{s'' \in G_i} T(s', a, s'')
\end{aligned}$$

Note that sums over t and s' could be interchanged because the summed terms are uniformely bounded by γ^t whose series is convergent. This would not be possible if we had $\gamma = 1$, which strengthens the use of *discounted* occupation measures in expressions converging to PCTL formula probabilities. Finally, since $\alpha(s) \neq 0$ if and only if $s \in \bigcap_{i=0}^{n} F_i \cap \overline{G_i}$, we have:

$$\begin{aligned}
\sum_{s \in S} \alpha(s) P_i^{\pi}(\gamma)(s) &= \sum_{\substack{s' \in F_i \cap \overline{G_i} \\ a \in A}} \sum_{t=0}^{+\infty} \gamma^t \sum_{s \in \bigcap_{i=0}^{n} F_i \cap \overline{G_i}} Pr(s_0 = s) \times \\
&\qquad Pr(s_t = s', a_t = a | s_0 = s, \pi) \sum_{s'' \in G_i} T(s', a, s'') \\
&= \sum_{\substack{s' \in F_i \cap \overline{G_i} \\ a \in A}} \sum_{t=0}^{+\infty} \gamma^t Pr(s_t = s', a_t = a | \alpha, \pi) \sum_{s'' \in G_i} T(s', a, s'') \\
&= \frac{1}{1-\gamma} \sum_{s' \in F_i \cap \overline{G_i}, a \in A} \mu_{\alpha}^{\pi}(s', a) \sum_{s'' \in G_i} T(s', a, s'') = P^{\pi}_{i,\alpha}(\gamma)
\end{aligned}$$

The final step is then obvious using Proposition 1. □

4.3 Iterative Linear Programming

For a given $0 < \gamma < 1$, Theorem 1 gives a linear expression of discounted PCTL formula probabilities in terms of occupation measures μ_{α}^{π}, which converge to the actual PCTL formula probabilities as γ tends to 1. In conjunction with the constraints on occupation measures (eq. 4) and the linear expression of the optimized value function as a function of occupation measures given earlier, we can

formulate the following $\mathbf{LP_\gamma}$ linear program, which has $|S||A|$ variables (vector $\mu \in S \times A$) and $n + |S|(1 + |A|)$ constraints:

$$\texttt{maximize:} \sum_{s\in S, a\in A} \mu(s,a) \sum_{s'\in S} T(s,a,s')R(s,a,s')$$

$$\texttt{subject to:}$$

$$\sum_{\substack{s'\in S\\ a\in A}} \mu(s',a)(\delta_s(s') - \gamma T(s',a,s)) = (1-\gamma)\alpha(s), \forall\, s \in S$$

$$\mu(s,a) \geqslant 0, \forall\, s \in S, a \in A$$

$$\sum_{s\in F_i\cap \overline{G_i}, a\in A} \mu(s,a) \sum_{s'\in G_i} T(s,a,s') \diamond_i (1-\gamma)p_i, \forall\, 1 \leqslant i \leqslant n$$

The idea is to solve many successive $\mathbf{LP_\gamma}$ problems with increasing values of γ, until we get a solution to $\mathbf{PCMDP\text{-}COP_\gamma}$, the path-constrained optimization problem formulated in section 3. Yet we need to precise the definition of solutions to $\mathbf{PCMDP\text{-}COP_\gamma}$, since its formulation, via the optimized objective function, depends on γ. Thus, the discount factor γ of the objective value function will depend on the number of iterations of $\mathbf{LP_\gamma}$ performed, instead of being chosen by the designer of the problem. However, in many applications, decision makers seek for the closest discount factor to 1 that provides sufficient long-term reasoning (γ impacts the long-term accumulation of rewards) while allowing for stable numerical stability and efficiency (which degrade as γ tends to 1). In other words, they would set γ to 1 if they could, except for applications where γ has special semantics. Therefore, we think that decision makers are interested in finding a solution of a given MDP problem for a *sufficiently high* discount factor, which gives rise to the following definition in the context of Path-Constrained MDPs.

Definition (γ-sufficient ϵ-optimality)**.** *Let $0 < \gamma < 1$ be some discount factor. A policy π is said to be a γ-sufficient ϵ-optimal policy if it is a solution of $\mathbf{PCMDP\text{-}COP_{\gamma'}}$, $\gamma' \geqslant \gamma$, such that $[P_{f_i}^{g_i}]_\infty^{\pi'}(\alpha) \diamond_i p_i \pm \epsilon$ for all $1 \leqslant i \leqslant n$ and all policies π'.*

Based on Theorem 1, which states that discounted PCTL formula probabilities tend to actual PCTL formula probabilities as γ tends to 1, we know that there exists $0 < \gamma_m < 1$ such that $\|P_{i,\alpha}^{\pi}(\gamma) - [P_{f_i}^{g_i}]_\infty^{\pi}(\alpha)\| < \epsilon$ for all $1 \leqslant i \leqslant n$ and $\gamma \geqslant \gamma_m$. Now, if we search for a γ_0-sufficient ϵ-policy, we can "simply" solve $\mathbf{LP_\gamma}$ with $\gamma = \max(\gamma_0, \gamma_m)$, because the objective function of $\mathbf{LP_\gamma}$ exactly matches the one of $\mathbf{PCMDP\text{-}COP_\gamma}$ for any γ. The following theorem proves that successively solving $\mathbf{LP_\gamma}$ with increasing γ values will eventually provide a solution to $\mathbf{PCMDP\text{-}COP_{\gamma_{final}}}$ if it is feasible, provided the inequalities (symbols $\diamond_i$) of constraints on PCTL formula probabilities are strict.

Theorem 2. *Assume a Path-Constrained MDP problem **PCMDP-COP**, such that $\diamond_i \in \{<, >\}$ for all $1 \leqslant i \leqslant n$, is feasible. Let be $0 < \gamma_0 < 1$ and $\epsilon > 0$. Then:*

(i) *It exists $0 < \gamma_1 < 1$ such that $\mathbf{LP_\gamma}$ is feasible for all $\gamma_1 \leqslant \gamma < 1$.*

(ii) *It exists $0 < \gamma_2 < 2$ such that $\|P_{i,\alpha}^{\pi}(\gamma) - [P_{f_i}^{g_i}]_\infty^{\pi}(\alpha)\| < \epsilon$ for all $1 \leqslant i \leqslant n$ and $\gamma \geqslant \gamma_2$ and all policy π.*

(iii) *Let $\mu^*(\gamma)$ be a solution of $\mathbf{LP_\gamma}$ with $\gamma = \max(\gamma_0, \gamma_1, \gamma_2)$, and $\pi^*(\gamma)$ be the stationary Markovian policy corresponding to $\mu^*(\gamma)$ as defined in eq. 3. Then, $\pi^*(\gamma)$ is a γ_0-sufficient ϵ-optimal policy of **PCMDP-COP**.*

Unfortunately, proving ϵ-optimality, i.e. item (ii) of Theorem 2, is very hard in the general case. In fact, eq. 6 shows that PCTL probability formulas have nearly the same mathematical structure as value functions of the total *undiscounted* reward criterion in MDPs, for which similar γ-convergence theoretical results have been reported, but without any algorithmic means to obtain ϵ-optimal value functions (for details, see Chapter 10 of [7]). However, for a given policy $\pi^*(\gamma)$ solution to $\mathbf{LP_\gamma}$ for a given γ, we can always exactly solve the linear systems of eq. 6 using $\pi^*(\gamma)$ (one system per PCTL constraint) and decide whether the computed PCTL formula probabilities satisfy all the constraints of **PCMDP-COP**.

This is the idea of Algorithm 1, named `ILP` for *Iterative Linear Programming*, which iterates over increasing γ discount factors, solving successive $\mathbf{LP_\gamma}$ problems (Line 3) until all the constraints of **PCMDP-COP** are satisfied (Lines 7 to 9). The initial discount factor is γ_0, given by the decision maker as a sufficient discount factor for the constrained value function to optimize. We increase discount factors γ using the following update formula (Line 10): $\gamma_{n+1} = (1-\gamma_0)\gamma_n + \gamma_0$, which ensures that $\lim_{n\to+\infty} \gamma_n = 1$, and which has a nice behavior for some γ_0 like 0.9 ($\gamma_1 = 0.99$, $\gamma_2 = 0.999$, etc.). Thanks to item (i) of Theorem 2, if **PCMDP-COP** is feasible, we will eventually find a feasible solution to $\mathbf{LP_\gamma}$ for some γ sufficiently close to 1. Therefore, iterations continue while $\mathbf{LP_\gamma}$ is not feasible or some PCTL formula probabilities are not satisfied ($flag$ is false in Line 12). In general, we do not know in advance if **PCMDP-COP** is feasible; if not, we have no guarantees to reach a discount factor γ such that $\mathbf{LP_\gamma}$ is feasible. This is why we stop iterations after a given number N of iterations in case $\mathbf{LP_\gamma}$ would not be feasible (or after PCTL formula probabilities are all satisfied).

Algorithm 1: Iterative Linear Programming (`ILP`)

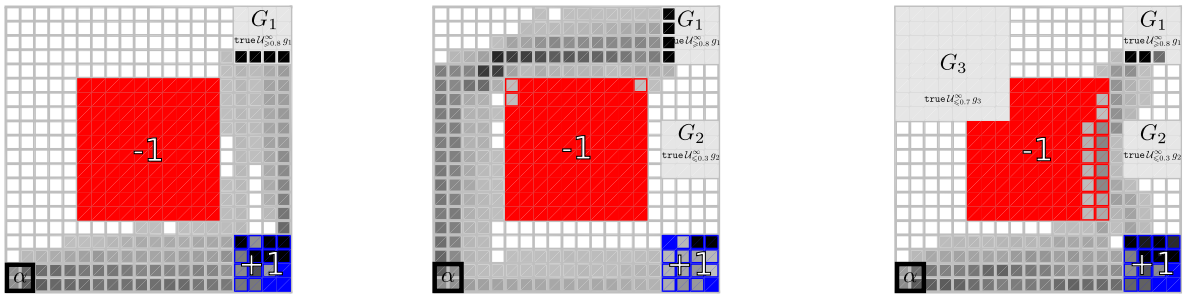
1 $\gamma \leftarrow \gamma_0$; $\quad flag \leftarrow$ `false`; $\quad iter \leftarrow 0$;
2 **repeat**
3 $\quad (\mu, V_\gamma^\pi) \leftarrow$ solve $\mathbf{LP_\gamma}$;
4 $\quad$ **if** ***LP*$_\gamma$** *is feasible* **then**
5 $\quad\quad \pi \leftarrow$ compute policy from μ using eq. 3;
6 $\quad\quad flag \leftarrow$ `true`;
7 $\quad\quad$ **for** $1 \leqslant i \leqslant n$ **do**
8 $\quad\quad\quad$ Compute $[P_{f_i}^{g_i}]_\infty^{\pi}(\alpha)$ by using eq. 2 or 6;
9 $\quad\quad\quad flag \leftarrow flag \wedge \left([P_{f_i}^{g_i}]_\infty^{\pi}(\alpha) \diamond_i p_i\right)$;
10 $\quad$ **if** $!flag$ **then** $\gamma \leftarrow (1-\gamma_0)\gamma + \gamma_0$;
11 $\quad iter \leftarrow iter + 1$;
12 **until** $flag \vee (iter > N)$;
13 **return** π, γ, V_γ^π, $\left([P_{f_i}^{g_i}]_\infty^{\pi}(\alpha)\right)_{1\leqslant i\leqslant n}$;

Finally, we mention the next theorem, following from Theorem 2, about completeness, optimality and termination properties of `ILP`.

Theorem 3. *`ILP` is complete and γ-sufficient ϵ-optimal if $\gamma < 1$ and $N = +\infty$ (unbounded number of iterations). With these settings, it terminates in finite time iff the **PCMDP-COP** problem is feasible. `ILP`'s solution is a stochastic Markovian policy computed using eq. 3 with the occupation measure that is solution of the last $\mathbf{LP_\gamma}$.*

5 EXPERIMENTAL RESULTS

In order to illustrate and evaluate our PC-MDP model, we tested navigation grid problems, where state trajectories of the optimal policy, impacted by PCTL constraints, are obvious and easily understandable. Our `ILP` algorithm can deal with far more complex PCTL formulas than the ones presented in these experiments, which were intentionally kept simple in order to ease understanding of the tradeoff between reward maximization and PCTL constraint satisfaction.



(a) 1 PCTL constraint: $\texttt{true}\,\mathcal{U}^{\infty}_{\geqslant 0.8}\, g_1$ (b) 2 PCTL constraints: (a) & $\texttt{true}\,\mathcal{U}^{\infty}_{\leqslant 0.3}\, g_2$ (c) 3 PCTL constraints: (b) & $\texttt{true}\,\mathcal{U}^{\infty}_{\leqslant 0.7}\, g_3$

Figure 1. Path-Constrained MDPs: impact of the number and the kind of PCTL constraints on the optimal policy

Figure 1 presents visual results of a same navigation problem (same reward structure) but with an increasing number of PCTL constraints. The agent starts at the bottom left corner. It can gather a +1 reward if it enters the bottom right corner, or pay a -1 reward if it enters the big central square. The first PCTL constraint (Fig. 1.a) consists in entering the top right corner with a probability *higher* than 0.8 (see PCTL formula in the figure). The second one (Fig. 1.b) aims at entering the middle right square with a probability *lower* than 0.3 (i.e. equivalent to *not* entering the square with sufficient probability). Finally, the third constraint (Fig. 1.c) consists also in entering the big top left square with a probability *lower* than 0.7. Experiments were conducted on a laptop equipped with a 2.30 GHz CPU and 2Gb of RAM. We used the COIN-OR-CLP simplex solver to solve $\mathbf{LP}_{\boldsymbol{\gamma}}$ problems, and the UMFPACK linear system solver to compute the (undiscounted) PCTL probabilities and check if they are satisfied.

In Fig. 1, we drew states (cells) visited by 100 stochastic simulations of the optimal policy: the darker the cell is, the most visited it is. With a single constraint, the strategy consists in going to the +1 reward area, then going to G_1 in order to satisfy the corresponding PCTL constraint. With an additional constraint (Fig. 1.b), the agent now prefers to go to the +1 reward area, then going to G_1 by going back the initial states and going around the central -1 reward, because this second PCTL constraint is satisfied if the agent does not enter it with sufficient probability. Finally, the third additional constraint (Fig. 1.c) is also satisfied if the agent does not enter it with sufficient probability; the agent chooses the first strategy (i.e. circling the central -1 area by the right), because it gets into a smaller portion of the central -1 area compared with the other strategy. We could wonder whether there exists an equivalent MDP, without PCTL constraints but with enriched reward structure, which would produce the same optimal policy. We think that there is no definitive answer to this question since: (i) the strategy with two constraints goes back to the initial states, so that standard MDP approaches would require to add some kind of trajectory history in the states; (ii) our policy is *guaranteed* to satisfy all PCTL constraints, whose mathematical properties are quite different from standard-MDP value functions.

Table 1 shows how `ILP` behaves when increasing the size of the grid ($|S| = size^2$), with always 3 constraints. We observed the final discount factor γ^* at convergence, the number of iterations $iter$, the total running time, the percentage of time used by all $\mathbf{LP}_{\boldsymbol{\gamma}}$s, and the percentage of time used to check the PCTL constraints (Line 8 of Alg. 1). As expected, the total time increases with the size of the grid. It is also true for γ^* and $iter$, which is rather intuitive because increasing the size of the grid requires more look-ahead. Moreover, we observe that most of the computation time is used to optimize the linear programs, and that checking PCTL formulas (recall that we compute exact PCTL formula probabilities for this) is negligible.

size	γ^*	$iter$	total time (s.)	% $\mathbf{LP}_{\boldsymbol{\gamma}}$	% **PCTL**
10	0.99	2	0.004145	71.15	23.55
25	0.999	3	0.115024	91.67	7.87
40	0.999	3	1.25937	97.44	2.5
50	0.999	3	19.2616	99.65	0.34
60	0.999	3	96.277	99.88	0.11
75	0.999	3	719.014	99.97	0.035

Table 1. Navigation problem: increasing grid sizes

6 CONCLUSION AND PERSPECTIVES

To the best of our knowledge, we presented one of the first theoretical framework to optimize MDP policies for the total average discounted reward criterion, under path-based constraints expressed in probability linear-time logic. We analyzed some mathematical properties of this new model, and proposed an iterative linear programming algorithm, named `ILP`, to solve PC-MDPs. While this study is rather academical, we think that the richness of the model will be of interest for many realistic applications, e.g. industrial ones where stochastic model-checking techniques are already used on a daily basis.

Many works remain to increase the class of problems that can be tackled: dealing with non-transient class sets F_i in PCTL formulas, taking into account different temporal horizons in the optimized value function and in the PCTL constraints, or even designing efficient forward-search algorithms to solve PC-MDPs.

REFERENCES

[1] Eitan Altman, *Constrained Markov decision processes*, Chapman & Hall/CRC, 1999.

[2] C. Courcoubetis and M. Yannakakis, 'Markov decision processes and regular events', *IEEE Transactions on Automatic Control*, **43**(10), 1399–1418, (October 1998).

[3] K. Etessami, M. Kwiatkowska, M. Y. Vardi, and M. Yannakakis, 'Multi-objective model checking of markov decision processes', in *Proc. of the 13th int. conf. on Tools and algorithms for the construction and analysis of systems*, p. 5065, Berlin, Heidelberg, (2007). Springer-Verlag.

[4] Hans Hansson and Bengt Jonsson, 'A logic for reasoning about time and reliability', *Formal Aspects of Computing*, **6**(5), 512–535, (1994).

[5] Andrey Kolobov, Mausam Mausam, Daniel S Weld, and Hector Geffner, 'Heuristic search for generalized stochastic shortest path MDPs', in *21st Int. Conference on Automated Planning and Scheduling*, (2011).

[6] M. Kwiatkowska, G. Norman, and D. Parker, 'Probabilistic symbolic model checking with PRISM: A hybrid approach', *Int. Journal on Software Tools for Technology Transfer (STTT)*, **6**(2), 128–142, (2004).

[7] Martin L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, Inc., New York, NY, USA, 1st edn., 1994.

[8] Florent Teichteil-Königsbuch, Guillaume Infantes, and Christel Seguin, 'Lazy forward-chaining methods for probabilistic model-checking', in *European Safety And Reliability Conference (ESREL 2011)*, Troyes, France, (2011). CRC Press.