One-Counter Markov Decision Processes

T. Brázdil * V. Brožek † K. Etessami † A. Kučera * D. Wojtczak‡†

Abstract

We study the computational complexity of some central analysis problems for One-Counter Markov Decision Processes (OC-MDPs), a class of finitely-presented, countable-state MDPs.

OC-MDPs extend finite-state MDPs with an unbounded counter. The counter can be incremented, decremented, or not changed during each state transition, and transitions may be enabled or not depending on both the current state and on whether the counter value is 0 or not. Some states are "random", from where the next transition is chosen according to a given probability distribution, while other states are "controlled", from where the next transition is chosen by the controller. Different objectives for the controller give rise to different computational problems, aimed at computing optimal achievable objective values and optimal strategies.

OC-MDPs are in fact equivalent to a controlled extension of (discrete-time) Quasi-Birth-Death processes (QBDs), a purely stochastic model heavily studied in queueing theory and applied probability. They can thus be viewed as a natural "adversarial" extension of a classic stochastic model. They can also be viewed as a natural probabilistic/controlled extension of classic one-counter automata. OC-MDPs also subsume (as a very restricted special case) a recently studied MDP model called "solvency games" that model a risk-averse gambling scenario.

Basic computational questions for OC-MDPs include "termination" questions and "limit" questions, such as the following: does the controller have a strategy to ensure that the counter (which may, for example, count the number of jobs in the queue) will hit value 0 (the empty queue) almost surely (a.s.)? Or that the counter will have lim sup value ∞, a.s.? Or, that it will hit value 0 in a selected terminal state, a.s.? Or, in case such properties are not satisfied almost surely, compute their optimal probability over all strategies.

We provide new upper and lower bounds on the complexity of such problems. Specifically, we show that several quantitative and almost-sure limit problems can be answered in polynomial time, and that almost-sure termination problems (without selection of desired terminal states) can also be answered in polynomial time. On the other hand, we show that the almost-sure termination problem with selected terminal states is PSPACE-hard and we provide an exponential time algorithm for this problem. We also characterize classes of strategies that suffice for optimality in several of these settings.

Our upper bounds combine a number of techniques from the theory of MDP reward models, the theory of random walks, and a variety of automata-theoretic methods.

1 Introduction

Markov Decision Processes (MDPs) are a standard model for stochastic dynamic optimization. They describe a system that exhibits both stochastic and controlled behavior. The system begins in some state and makes a sequence of state transitions; depending on the state, either the controller gets to choose from among possible transitions, or there is a probability distribution over possible transitions.[1] Fixing a *strategy* for the controller determines a probability space of (potentially infinite) runs, or trajectories, of the MDP. The controller's goal is to optimize the (expected) value of some objective function, which may be a function of the entire trajectory. Two fundamental computational questions that arise are "*what is the optimal value that the controller can achieve?*" and "*what strategies achieve this?*". For finite-state MDPs, such questions have been studied for many objectives and there is a large literature on both the complexity of central questions as well as on methods that work well in practice, such as value iteration and policy iteration (see, e.g., [23]).

Many important stochastic models are, however, not finite-state, but are finitely-presented and describe an infinite-state underlying stochastic process. Classic examples include branching processes, birth-death processes, and many others. Computational questions for such purely stochastic models have also been studied for a long time. A model that is of direct relevance to this paper is the Quasi-Birth-Death process (QBD), a generalization of birth-death processes that has been heavily studied in queueing theory and applied probability (see, e.g., the books [21, 20, 3]).

*xbrazdil@fi.muni.cz, tony@fi.muni.cz, Faculty of Informatics, Masaryk University

†kousha@inf.ed.ac.uk, vaclav.brozek@gmail.com, School of Informatics, University of Edinburgh

‡CWI, Amsterdam, D.K.Wojtczak@cwi.nl

[1] Our focus is on discrete state spaces, and discrete-time MDPs. In some presentations of such MDPs, probabilistic and controlled transitions are combined into one: each transition entails a controller move followed by a probabilistic move. The two presentations are equivalent.

Intuitively, a QBD describes an unbounded queue, using a counter to count the number of jobs in the queue, and such that the queue can be in one of a bounded number of distinct "modes" or "states". Stochastic transitions can add or remove jobs from the queue and can also transition the queue from one state to another. QBDs are in general studied as continuous-time processes, but many of their key analyses (including both steady-state and transient analyses) amount to analysis of their underlying embedded discrete-time QBD (see, e.g., [20]). An equivalent way to view discrete-time QBDs is as a probabilistic extension of classic *one-counter automata* (see, e.g, [26]), which extend finite-state automata with an unbounded counter. The counter can be incremented, decremented, or remain unchanged during state transitions, and transitions may be enabled or not depending on both the current state and on whether the counter value is 0 or not. In *probabilistic* one-counter automata (i.e., QBDs), from every state the next transition is chosen according to a probability distribution depending on that state. (See [10] for much more information on the relation between QBDs and other models.)

In this paper we study *One-Counter Markov Decision Processes* (OC-MDPs), which extend discrete-time QBDs with a controller. An OC-MDP has a finite set of states: some states are *random*, from where the next transition is chosen according to a given probability distribution, and other states are *controlled*, from where the next transition is chosen by the controller. Again, transitions can change the state and can also change the value of the (unbounded) counter by at most 1. Different objectives for the controller give rise to different computational problems for OC-MDPs, aimed at optimizing those objectives.

Motivation for studying OC-MDPs comes from several different directions. Firstly, it is very natural, both in queueing theory and in other contexts, to consider an "adversarial" extension of stochastic models like QBDs, so that stochastic assumptions can sometimes be replaced by "worst-case" or "best-case" assumptions. For example, under stochastic assumptions about arrivals, we may wish to know whether there exists a "best-case" control of the queue under which the queue will almost surely become empty (such questions are of course related to the stability of the queue), or we may ask if we can do this with at least a given probability. Such questions are similar in spirit to questions asked in the rich literature on "adversarial queueing theory" (see, e.g., [4]), although this is a somewhat different setting. These considerations lead naturally to the extension of QBDs with control, and thus to OC-MDPs. Indeed, MDP variants of QBDs have already been studied in the stochastic modeling literature, see [27, 19]. However, in order to keep their analyses tractable, these works take the drastic approach of cutting off the value of the counter (i.e., size of the queue) at some arbitrary finite value N, effectively adding dead-end absorbing states at values higher than N. This restricts the model to a finite-state "approximation". However, cutting off the counter value can radically alter the behavior of the model, even for purely probabilistic QBDs, and it is easy to construct examples that exhibit this (see the full version [5] for simple examples). Thus the existing work in the QBD literature on MDPs does not establish any results about the computational complexity, or even decidability, of basic analysis problems for general OC-MDPs.

OC-MDPs also subsume another recently studied infinite-state MDP model called *solvency games* [1], which amount to a very limited subclass of OC-MDPs. Solvency games model a risk-averse "gambler" (or "investor"). The gambler has an initial pot of money, given by a positive integer, n. He/she then has to choose repeatedly from among a finite set of possible gambles, each of which has an associated random gain/loss given by a finite-support probability distribution over the integers. Berger et. al. [1] study the gambler objective of minimizing the probability of going bankrupt. One can of course study the same basic repeated gambling model under a variety of other objectives, and many such objectives have been studied. It is not hard to see that all such repeated gambling models constitute special cases of OC-MDPs. The counter in an OC-MDP can keep track of the gambler's wealth. Although, by definition, OC-MDPs can only increment or decrement the counter by one in each state transition, it is easy to augment any finite change to the counter value by using auxiliary states and incrementing or decrementing the counter by one at a time. Similarly, with an OC-MDP one can easily augment any choice over finite-support probability distribution on integers, each of which defines the random change to the counter corresponding to a particular gamble. [1] showed that if the solvency game satisfies several additional restrictive technical conditions, then one can characterize the optimal strategies for minimizing the probability of bankruptcy (as a kind of "ultimately memoryless" strategy) and compute them using linear programming. They did not however establish any results for general, unrestricted, solvency games. They conclude with the following remark: "It is clear that our results are at best a sketch of basic elements of a larger theory". We believe OC-MDPs constitute an appropriate larger framework within which to study algorithmic questions not just for solvency games, but for various more general infinite-state MDP models that employ a counter. In Proposition 4.1, we show that all *qualitative* questions about (unrestricted) solvency games, namely whether the gambler has a strategy to not go bankrupt with probability > 0, $= 1$, $= 0$, < 1, can be answered in polynomial time.

Our goal it to study the computational complexity of central analysis problems for OC-MDPs. Key quantities associated with discrete-time QBDs, which can be used to derive many other useful quantities, are "termination

probabilities" (also known as their "G matrix"). These are the probabilities that, starting from a given state, with counter value 1, we will eventually reach counter value 0 for the first time in some other given state. The complexity of computing termination probabilities for QBDs is already an intriguing problem, and many numerical methods have been devised for it. A recent result in [10] shows that these probabilities can be approximated in time polynomial in the size of the QBD, in the unit-cost RAM model of computation, using a variant of Newton's method, but that deciding , e.g., whether a termination probability is $\geq p$ for a given rational $p \in (0, 1)$ in the standard Turing model is at least as hard as a long standing open problem in exact numerical computation, namely the square-root sum problem, which is not even known to be in NP nor the polynomial-time hierarchy. (See [10] for more information.)

We study OC-MDPs under related objectives, in particular, the objective of maximizing termination probability, and of maximizing the probability of termination in a particular subset of the states (the latter problem is considerably harder, as we shall see). Partly as a stepping stone toward these objectives, but also for its own intrinsic interest, we also consider OC-MDPs without boundary, meaning where the counter can take on both positive and negative values, and we study the objective of optimizing the probability that the lim sup value is $= \infty$ (or, by symmetry, that the lim inf is $= -\infty$). The boundaryless model is related, in a rather subtle way, to the well-studied model of finite-state MDPs with limiting average reward objectives (see, e.g., [23]). This connection enables us to exploit recent results for finite-state MDPs ([15]), and classic facts in the theory of 1-dimensional random walks and sums of i.i.d. random variables, to analyze the boundaryless case of OC-MDPs. We then use these analyses as crucial building blocks for the analysis of optimal termination probabilities in the case of OC-MDPs with boundary. Our main results are the following:

1. For boundaryless OC-MDPs, where the objective of the controller is to maximize the probability that the lim sup (lim inf) of the counter value in the run (the trajectory) is ∞ ($-\infty$), the situation is as good as we could hope; namely, we show:
   (a) The optimal probability is a rational value that is polynomial-time computable.
   (b) There exist deterministic optimal strategies that are both "*counter-oblivious*" and *memoryless* (we shall call these CMD strategies), meaning the choice of the next transition depends only on the current state and neither on the history, nor on the current counter value.
   Furthermore, such an optimal strategy can be computed in polynomial time.
2. For OC-MDPs with boundary, where the objective is to maximize the probability that, starting in some state and with counter value 1, we eventually *terminate* (reach counter value 0) *in any state*, we have:
   (a) In general the optimal (supremum) probability can be an irrational value, and this is so already in the case of QBDs where there is no controller, see [10].
   (b) It is decidable in polynomial time whether the optimal probability is 1.
   (c) There is a CMD strategy such that starting from every state with value 1, using that strategy we terminate almost surely.
   (Optimal CMD strategies need not exist starting from states where the optimal probability is < 1.)
3. For OC-MDPs with boundary, where the objective is to maximize the probability that, starting from a given state and counter value 1, we terminate in a *selected* subset of states F (i.e., reach counter value 0 for the first time in one of these selected states), we know the following:
   (a) There need not exist any optimal strategy, even when the supremum probability of termination in selected states is 1 (i.e., only ϵ-optimal strategies may exist).
   (b) Even deciding whether there is an optimal strategy which ensures probability 1 termination in the selected states is PSPACE-hard.
   (c) We provide an exponential time algorithm to determine whether there is a strategy using which the probability of termination in the selected states is 1, starting at a given state and counter value.

Our proofs employ techniques from several areas: from the theory of finite-state MDP reward models (including some recent results), from the theory of 1-dimensional random walks and sums of i.i.d. random variables, and a variety of automata-theoretic methods (e.g., pumping arguments, decomposition arguments, etc.).

Our results leave open many interesting questions about OC-MDPs. For example, we do not know whether the following problem is decidable: given an OC-MDP and a rational probability $p \in (0, 1)$, decide whether the optimal probability of termination (in any state) is greater than p. Other open questions pertain to OC-MDPs where the objective is to minimize termination probabilities. We view this paper as laying the basic foundations for the algorithmic analysis of OC-MDPs, and we feel that answering some of the remaining open questions will reveal an even richer underlying theory.

Related work. A more general MDP model that strictly subsumes OC-MDPs, called *Recursive Markov Decision Processes* (RMDPs) was studied in [11, 12]. These are equivalent to MDPs whose state transition structure is that of a general pushdown automaton. Problems such as deciding whether there is a strategy that yields termination probability 1, or even approximating the maximum probability within any non-trivial additive factor, were shown to be undecidable for general RMDPs in [11]. For the restricted class of 1-exit RMDPs (which correspond in a precise sense to MDP versions of multi-type branching processes, stochastic context-free grammars, and a related model called pBPAs), [11] showed quantitative problems for optimal termination probability are decidable in PSPACE, and [12] showed that deciding whether the optimal termination probability is 1 can be done in P-time. In [6] this was extended further to answer certain qualitative almost-sure reachability questions for 1-exit RMDPs in P-time. 1-exit RMDPs are however incompatible with OC-MDPs (which actually correspond to 1-box RMDPs). In [10], quantiative termination problems for purely stochastic QBDs were studied. The references in these cited papers point to earlier related literature, in particular on probabilistic Pushdown Systems and Recursive Markov chains. There is a substantial literature on numerical algorithms for analysis of QBDs and related purely stochastic models (see [21, 20, 3]). In that literature one can find results related to qualitative questions, like whether the termination probability for a given QBD is 1. Specifically, it is known that for an *irreducible* QBD, i.e., a QBD in which from every configuration (counter value and state) one can reach every other configuration with non-zero probability, whether the underlying Markov chain is recurrent boils down to steady-state analysis of induced finite-state chains over states of the QBD, and in particular on whether the expected one-step change in the counter value in steady state is ≤ 0 (see, e.g., Chapter 7 of [20] for a proof). However, these results crucially assume the QBD is irreducible. They do not directly yield an algorithm for deciding, for general QBDs, whether the probability of termination is 1 starting from a given state and counter value 1. Thus, our results for OC-MDPs yield new results even for purely stochastic QBDs without controller.

2 Basic definitions

We fix some notation. We use $\mathbb{Z}, \mathbb{N}, \mathbb{N}_0$ to denote the integers, positive integers, and non-negative integers, respectively. We use standard notation for intervals, e.g., $(0, 1]$ denotes $\{x \in \mathbb{R} \mid 0 < x \leq 1\}$. The set of finite words over an alphabet Σ is denoted Σ^*, and the set of infinite words over Σ is denoted Σ^ω. Σ^+ denotes $\Sigma^* \setminus \{\varepsilon\}$ where ε is the empty word. The length of a given $w \in \Sigma^* \cup \Sigma^\omega$ is denoted $|w|$, where the length of an infinite word is ∞. Given a word (finite or infinite) over Σ, the individual letters of w are denoted $w(0), w(1), \cdots$ (so indexing begins at 0). For a word w, we denote by $w{\downarrow}n$ the prefix $w(0)\cdots w(n-1)$ of w. Let $\mathcal{V} = (V, \to)$ where V is a non-empty set and $\to \subseteq V \times V$ a *total* relation (i.e., for every $v \in V$ there is some $u \in V$ such that $v \to u$). The reflexive transitive closure of $\to$ is denoted $\to^*$. A *path* in $\mathcal{V}$ is a finite or infinite word $w \in V^+ \cup V^\omega$ such that $w(i-1) \to w(i)$ for every $1 \leq i < |w|$. A *run* in $\mathcal{V}$ is an infinite path in V. The set of all runs in $\mathcal{V}$ is denoted $Run_{\mathcal{V}}$. The set of runs in $\mathcal{V}$ that start with a given finite path w is denoted $Run_{\mathcal{V}}(w)$.

We assume familiarity with basic notions of probability, e.g., a σ-field, $\mathcal{F}$, over a set Ω, and a probability measure $\mathcal{P} : \mathcal{F} \mapsto [0, 1]$, together define a *probability space* $(\Omega, \mathcal{F}, \mathcal{P})$. As usual, a *probability distribution* over a finite or countably infinite set X is a function $f : X \to [0, 1]$ such that $\sum_{x \in X} f(x) = 1$. We call f *positive* if $f(x) > 0$ for every $x \in X$, and *rational* if $f(x) \in \mathbb{Q}$ for every $x \in X$.

For our purposes, a *Markov chain* is a triple $\mathcal{M} = (S, \to, Prob)$ where S is a finite or countably infinite set of *states*, $\to \subseteq S \times S$ is a total *transition relation*, and *Prob* is a function that assigns to each state $s \in S$ a positive probability distribution over the outgoing transitions of s. As usual, we write $s \xrightarrow{x} t$ when $s \to t$ and x is the probability of $s \to t$. To every $s \in S$ we associate the probability space $(Run_{\mathcal{M}}(s), \mathcal{F}, \mathcal{P})$ of runs starting at s, where $\mathcal{F}$ is the σ-field generated by all *basic cylinders*, $Run_{\mathcal{M}}(w)$, where w is a finite path starting with s, and $\mathcal{P} : \mathcal{F} \to [0, 1]$ is the unique probability measure such that $\mathcal{P}(Run_{\mathcal{M}}(w)) = \prod_{i=1}^{|w|-1} x_i$ where $w(i-1) \xrightarrow{x_i} w(i)$ for every $1 \leq i < |w|$. If $|w| = 1$, we put $\mathcal{P}(Run_{\mathcal{M}}(w)) = 1$.

Definition 2.1. *A* **Markov decision process (MDP)** *is a tuple $\mathcal{D} = (V, \hookrightarrow, (V_N, V_P), Prob)$, where V is a finite or countable set of* vertices*, $\hookrightarrow \subseteq V \times V$ is a total* transition relation*, (V_N, V_P) is a partition of V into* non-deterministic *(or "controlled") and* probabilistic *vertices, and Prob is a* probability assignment *which to each $v \in V_P$ assigns a rational probability distribution on its set of outgoing transitions.*

A *strategy* is a function σ which to each $wv \in V^* V_N$ assigns a probability distribution on the set of outgoing transitions of v. We say that a strategy σ is *memoryless (M)* if $\sigma(wv)$ depends only on the last vertex v, and *deterministic (D)* if $\sigma(wv)$ is a Dirac distribution (assigns probability 1 to some transition) for each $wv \in V^* V_N$. When σ is D, we write $\sigma(wv) = v'$ instead of $\sigma(wv)(v, v') = 1$. For a memoryless deterministic (MD) strategy σ, we write $\sigma(v) = v'$ instead of $\sigma(wv)(v, v') = 1$. Strategies that are not necessarily memoryless (respectively, deterministic) are called *history-dependent (H)* (respectively, *randomized (R)*). We use *HR* to denote the set of all (i.e., *H* and *R*) strategies.

Each strategy σ determines a unique Markov chain $\mathcal{D}(\sigma)$ for which V^+ is the set of states, and $wu \xrightarrow{x} wuu'$ iff $u \hookrightarrow u'$ *and* one of the following conditions holds: (1) $u \in V_P$

and $Prob(u \hookrightarrow u') = x$, or (2) $u \in V_N$ and $\sigma(wu)$ assigns x to the transition $u \hookrightarrow u'$. To every $w \in Run_{\mathcal{D}(\sigma)}$ we associate the corresponding run $w_{\mathcal{D}} \in Run_{\mathcal{D}}$ where $w_{\mathcal{D}}(i)$ is the vertex currently *visited* by $w(i)$, i.e., the last element of $w(i)$ (note $w(i) \in V^+$).

For our purposes in this paper, an *objective*[2] is a set $O \subseteq Run_{\mathcal{D}}$ (in situations when the underlying MDP $\mathcal{D}$ is not clear from the context, we write $O_{\mathcal{D}}$ instead of O). For every strategy σ, let O^σ be the set of all $w \in Run_{\mathcal{D}(\sigma)}$ such that $w_{\mathcal{D}} \in O$. Further, for every $v \in V$ we use $O^\sigma(v)$ to denote the set of all $w \in O^\sigma$ which start at v. We say that O is *measurable* if $O^\sigma(v)$ is measurable for all σ and v. For a measurable objective O and a vertex v, the *O-value in v* is defined as follows: $Val^O(v) = \sup_{\sigma \in HR} \mathcal{P}(O^\sigma(v))$. We say that a strategy σ is *O-optimal* starting at a given vertex v if $\mathcal{P}(O^\sigma(v)) = Val^O(v)$. We say σ is *O-optimal*, if it is optimal starting at every vertex. An important objective for us is *reachability*. For every set $T \subseteq V$ of *target vertices*, we define the objective $Reach_T = \{w \in Run_{\mathcal{D}} \mid \exists i \in \mathbb{N}_0 \text{ s.t. } w(i) \in T\}$.

Definition 2.2. *A* **one-counter MDP (OC-MDP)** *is a tuple, $\mathcal{A} = (Q, \delta^{=0}, \delta^{>0}, (Q_N, Q_P), P^{=0}, P^{>0})$, where*

- *Q is a finite set of* states*, partitioned into* non-deterministic*, Q_N, and* probabilistic*, Q_P, states.*
- *$\delta^{>0} \subseteq Q \times \{-1, 0, 1\} \times Q$ and $\delta^{=0} \subseteq Q \times \{0, 1\} \times Q$ are the sets of* positive *and* zero rules *(transitions) such that each $p \in Q$ has an outgoing positive rule and an outgoing zero rule;*
- *$P^{>0}$ and $P^{=0}$ are* probability assignments*: both assign to each $p \in Q_P$, a positive rational probability distribution over the outgoing transitions in $\delta^{>0}$ and $\delta^{=0}$, respectively, of p.*

Each OC-MDP, $\mathcal{A}$, naturally determines an infinite-state MDP with or without a boundary, depending on whether zero testing is taken into account or not. Formally, we define MDPs $\mathcal{D}^{\rightarrow}_{\mathcal{A}}$ and $\mathcal{D}^{\leftrightarrow}_{\mathcal{A}}$ as follows:

- $\mathcal{D}^{\rightarrow}_{\mathcal{A}} = (Q \times \mathbb{N}_0, \mapsto, (Q_N \times \mathbb{N}_0, Q_P \times \mathbb{N}_0), Prob)$. Here for all $p, q \in Q$ and $j \in \mathbb{N}_0$ we have that $p(0) \mapsto q(j)$ iff $(p, j, q) \in \delta^{=0}$. If $p \in Q_P$, then the probability of $p(0) \mapsto q(j)$ is $P^{=0}(p, j, q)$. Further for all $p, q \in Q$, $i \in \mathbb{N}$, and $j \in \mathbb{N}_0$ we have that $p(i) \mapsto q(j)$ iff $(p, j{-}i, q) \in \delta^{>0}$. If $p \in Q_P$, then the probability of $p(i) \mapsto q(j)$ is $P^{>0}(p, j{-}i, q)$.
- $\mathcal{D}^{\leftrightarrow}_{\mathcal{A}} = (Q \times \mathbb{Z}, \mapsto, (Q_N \times \mathbb{Z}, Q_P \times \mathbb{Z}), Prob)$, where for all $p, q \in Q$ and $i, j \in \mathbb{Z}$ we have that $p(i) \mapsto q(j)$ iff $(p, j{-}i, q) \in \delta^{>0}$. If $p \in Q_P$, then the probability of $p(i) \mapsto q(j)$ is $P^{>0}(p, j{-}i, q)$.

Since the MDPs $\mathcal{D}^{\rightarrow}_{\mathcal{A}}$ and $\mathcal{D}^{\leftrightarrow}_{\mathcal{A}}$ have infinitely many vertices, even *MD* strategies are not necessarily finitely representable. But the objectives we consider are often achievable with strategies that use only finite information about the counter or even ignore the counter value. We call a strategy, σ, in $\mathcal{D}^{\rightarrow}_{\mathcal{A}}$ or $\mathcal{D}^{\leftrightarrow}_{\mathcal{A}}$, *counter-oblivious*-MD (denoted *CMD*) if there is a *selector*, $f : Q \to \delta^{>0}$ (which selects a transition out of each state) so that at any configuration $p(n) \in Q \times \mathbb{N}$, σ chooses transition $f(p)$ with probability 1 (ignoring history and n).

3 OC-MDPs Without Boundary

In this section we study the objective "Cover Negative" (CN), which says that values of the counter during the run should cover arbitrarily low negative numbers in $\mathbb{Z}$ (i.e., that the lim inf counter value in the run is $= -\infty$). Our goal is to prove Theorem 3.1, below. (All missing proofs missing can be found in the full version [5].)

Definition 3.1. *Let $\mathcal{A}$ be a OC-MDP. We use $CN_{\mathcal{A}}$ to denote the set of all runs $w \in Run_{\mathcal{D}^{\rightarrow}_{\mathcal{A}}}$ such that for every $n \in \mathbb{Z}$ the run w visits a configuration $p(i)$ for some $p \in Q$ and $i \leq n$.*

Theorem 3.1. *Given a OC-MDP, $\mathcal{A}$, there is a $CN_{\mathcal{A}}$-optimal CMD strategy for it, which is computable in polynomial time. Moreover, $Val^{CN_{\mathcal{A}}}$ is rational and computable in polynomial time.*

We prove this via a sequence of reductions to problems for finite-state MDPs with and without *rewards*. For us a *MDP with reward*, $\mathcal{D}$, is equipped with a reward function $r : V \to \{-1, 0, 1\}$. For $v = v_0 \cdots v_n \in V^+$, we define the reward $r(v) := \sum_{i=0}^{n} r(v_i)$.

Definition 3.2. *We denote by CN the set of all $w \in Run_{\mathcal{D}}$ satisfying $\liminf_{n\to\infty} r(w{\downarrow}n) = -\infty$. We further denote by MP the set of all runs $w \in Run_{\mathcal{D}}$ such that $\lim_{n\to\infty} \frac{r(w{\downarrow}n)}{n}$ exists and $\lim_{n\to\infty} \frac{r(w{\downarrow}n)}{n} \leq 0$.*[3]

The following is a consequence of a theorem by Gimbert ([15, Theorem 1]).

Lemma 3.1. (cf. [15]) *For finite-state MDPs with rewards, there always exists a CN-optimal MD strategy.*

This follows from [15, Theorem 1] because (the characteristic function of) the objective *CN* is both *prefix-independent* and *submixing*, and Gimbert's theorem shows

[2] In general, objectives can be arbitrary Borel measurable functions of trajectories, for which we want to optimize expected value. We only consider objectives that are characteristic functions of a measurable set of trajectories.

[3] "MP" stands for "(non-positive) Mean Payoff".

that these two criteria are sufficient conditions for the existence of an optimal MD strategy. Briefly, prefix-independence means that only the tail of an infinite run is relevant for determining its payoff, and submixing means that if any given run is cut into infinitely many finite segments and these segments are partitioned into two parts to make two infinite "runs", then the maximum payoff among the two created runs is at least as large as the payoff of the original run. (See the full version for [5] for details.) Lemma 3.2 below shows that for OC-MDPs there is also always a $CN_{\mathcal{A}}$-optimal CMD strategy, which is a fairly simple consequence of the same fact for finite-state MDPs with rewards. We now define a sequence of problems which we shall use in reductions for establishing Theorem 3.1:

OC-MDP-CN:
Input: OC-MDP, $\mathcal{A}$, and $z \in \mathbb{Z}$.
Output: a $CN_{\mathcal{A}}$-optimal CMD strategy for $\mathcal{A}$, and $Val^{CN_{\mathcal{A}}}(p(z))$, for every $p \in Q$.

MDP-CN:
Input: finite-state MDP, $\mathcal{D}$, with reward function r.
Output: a CN-optimal MD strategy for $\mathcal{D}$, and $Val^{CN}(v)$, for every vertex v of $\mathcal{D}$.

MDP-CN-qual:
Input: finite-state MDP, $\mathcal{D}$, with reward function r.
Output: set $A = \{v \mid Val^{CN}(v) = 1\}$, and a MD strategy σ which is CN-optimal starting at every $v \in A$.

MDP-MP-qual:
Input: finite-state MDP, $\mathcal{D}$, with reward function r.
Output: set $A = \{v \mid \exists \sigma_v \in MD : \mathcal{P}(MP^{\sigma_v}(v)) = 1\}$, a $\bar{\sigma} \in MD$ such that $\forall v \in A : \mathcal{P}(MP^{\bar{\sigma}}(v)) = 1$.[4]

Proposition 3.1. *1. There exist the following polynomial-time (Turing) reductions:*

$$\textbf{\textit{OC-MDP-CN}} \leq_P \textbf{\textit{MDP-CN}} \leq_P \textbf{\textit{MDP-CN-qual}} \leq_P \textbf{\textit{MDP-MP-qual}}$$

2. The problem ***MDP-MP-qual*** *can be solved in polynomial time.*

The following lemma establishes both the first reduction of Proposition 3.1, part 1, and the existence of $CN_{\mathcal{A}}$-optimal CMD strategies for OC-MDPs.

Lemma 3.2. *Given a OC-MDP, $\mathcal{A}$, there is a finite-state MDP with rewards, $\mathcal{D}$, computable in polynomial time from $\mathcal{A}$, such that the set of vertices of $\mathcal{D}$ contains Q and for every $p \in Q$, $i \in \mathbb{Z}$ we have that $Val^{CN_{\mathcal{A}}}(p(i)) = Val^{CN}(p)$.*

[4]The existence of strategy $\bar{\sigma}$ is a consequence of the correctness proof for procedure `Qual-MP` which can be found in the full version [5].

Procedure `Solve-CN(`$\mathcal{D}$`,`r`)`

Data: A MDP $\mathcal{D}$ with reward r.
Result: Compute the vector $\left(Val^{CN}(v)\right)_{v \in V}$, and a CN-optimal MD strategy, σ.

1 $(A, \tau) \leftarrow$ `Qual-CN(`$\mathcal{D}$`,`r`)`
2 $(\sigma_R, (val_v)_{v \in V}) \leftarrow$ `Max-Reach(`$\mathcal{D}$`,`A`)`
3 **for** *every* $v \in V_N$ **do if** $v \in A$ **then** $\sigma(v) \leftarrow \tau(v)$ **else** $\sigma(v) \leftarrow \sigma_R(v)$
4 **return** $(val_v)_{v \in V}$, σ

Moreover, for a MD strategy σ in $\mathcal{D}$, let σ' be the CMD strategy in $\mathcal{D}^{\leftrightarrow}_{\mathcal{A}}$ with a selector f defined by $f(p) = \sigma(p)$. Then for each $p(i) \in Q \times \mathbb{Z}$, $\mathcal{P}(CN^{\sigma'}_{\mathcal{A}}(p(i))) = \mathcal{P}(CN^{\sigma}(p))$.

The proof of Lemma 3.2 is very easy. Indeed, in this boundaryless setting, the objective of making the OC-MDP's counter value hit arbitrarily small negative numbers is basically equivalent to making the accumulated total reward hit arbitrarily small negative values in the underlying finite-state MDP, where the one-step reward is defined to be the one-step change in the counter value. (The only minor difference is that we have defined rewards on states rather than transitions, but this can be handled easily using auxiliary states.)

By Lemma 3.1 and 3.2, we only have to consider MD strategies. Dealing with MD strategies simplifies notation. Although as defined the Markov chain $\mathcal{D}(\sigma)$ has infinitely many states, for a finite-state MDP $\mathcal{D} = (V, \hookrightarrow, (V_N, V_P), Prob)$ and a MD strategy σ we can clearly replace $\mathcal{D}(\sigma)$ with a finite-state Markov chain $\mathcal{D}\langle\sigma\rangle$ where V is the set of states, and $u \xrightarrow{x} u'$ iff $u \xrightarrow{x} uu'$ in $\mathcal{D}(\sigma)$. This only changes notation since for every $u \in V$ there is an isomorphism between the probability spaces $Run_{\mathcal{D}(\sigma)}(u)$ and $Run_{\mathcal{D}\langle\sigma\rangle}(u)$ given by the bijection of runs which maps run w to $w_{\mathcal{D}}$, see the definition of $\mathcal{D}(\sigma)$ in Section 2.

To finish the proof of Theorem 3.1 we have to: (1.) provide the last two reductions from Proposition 3.1, part 1, prove that Val^{CN} is always rational, and prove Proposition 3.1, part 2. We do each of these in separate subsections.

3.1 Reduction to Qualitative CN.

The key to establishing the reduction **MDP-CN** $\leq_P$ **MDP-CN-qual** is the following:

Proposition 3.2. *Let $A := \{v \in V \mid Val^{CN}(v) = 1\}$. Then for all $u \in V$ we have:*

$$Val^{CN}(u) = \max_{\tau \in MD} \mathcal{P}(Reach^{\tau}_{A}(u)) = \sup_{\tau \in HR} \mathcal{P}(Reach^{\tau}_{A}(u))$$

In other words, the optimal probability of CN in a finite-state MDP with reward, is precisely the optimal probability of reaching some vertex from which there is a strategy to achieve CN with probability 1. The proof of this proposition can be sketched as follows. The fact that optimal MD strategies exist for reachability objectives, i.e.,

that $\max_{\tau \in MD} \mathcal{P}(Reach_A^\tau(u)) = \sup_{\tau \in HR} \mathcal{P}(Reach_A^\tau(u))$, follows from well established facts about MDPs with reachability objectives, see, e.g., [23, Section 7.2.7] or [8]. Clearly $\max_{\tau \in MD} \mathcal{P}(Reach_A^\tau(u)) \leq Val^{CN}(u)$, because once we reach a vertex from which we can achieve CN with probability 1, we can switch to the appropriate strategy for achieving that. For the opposite direction, let us pick a *CN*-optimal MD strategy σ, which we know exists by Lemma 3.1. Consider the resulting finite-state Markov chain $\mathcal{D}\langle\sigma\rangle$ with states V. There will be some bottom strongly connected components (BSCCs) of the underlying directed graph of this resulting Markov chain. The crucial observation is this: consider some BSCC, C, and a node u in that BSCC. Let X_u denote the random variable that describes the change in reward value (counter value) between consecutive visits to state u in a random walk on the Markov chain $\mathcal{D}\langle\sigma\rangle$. (Note that the probability of not revisiting u is 0.) Now we can basically view the change in reward value during an entire infinite run starting at u as a sum of i.i.d. random variables, $S_n = \sum_{i=1}^n X_i$, where each X_i has the same distribution as X_u. It follows from classic facts about random walks on the real line, i.e., sums of i.i.d. random variables, that the probability $\mathcal{P}(\liminf_{n\to\infty} S_n = -\infty)$ is either 0 or 1, and that probability 1 happen precisely when $E[X_u] \leq 0$ and X_u is not trivial, i.e., is not identically 0 (in which case clearly the probability would be 0). See, e.g, [7, Theorem 8.2.5 and Theorem 8.3.4] for a proof of these facts. Thus, since σ was optimal for CN, and since with probability 1 any run on $\mathcal{D}\langle\sigma\rangle$ eventually enters one of the BSCCs, optimizing the probability of satisfying CN starting at any vertex v amounts to optimizing the probability of reaching one of the BSCCs from which σ achieves CN with probability 1.

The reduction **MDP-CN** $\leq_P$ **MDP-CN-qual** is described in procedure `Solve-CN`. Its correctness follows from Proposition 3.2. Once the set A of vertices with $Val^{CN} = 1$, and a corresponding *CN*-optimal strategy, are both computed (line 1, which calls the subroutine `Qual-CN` for solving **MDP-CN-qual**), solving **MDP-CN** amounts to computing an MD strategy for maximizing the probability of reaching a vertex in A, and computing the respective reachability probabilities. This is done on line 2 by calling procedure `Max-Reach`. It is well known that `Max-Reach` can be implemented in polynomial time via linear programming: both an optimal strategy and the associated optimal (rational) probabilities can be obtained by solving suitable linear programs (see, e.g., [8] or [23, Section 7.2.7]). Thus the running time of `Solve-CN`, excluding the running time of `Qual-CN`, is polynomial. Moreover, the optimal values are rational, so Lemma 3.2 implies that $Val^{CN,\mathcal{A}}$ is also rational.

3.2 Reduction to Qualitative *MP*. The reduction **MDP-CN-qual** $\leq_P$ **MDP-MP-qual** is described in procedure `Qual-CN`. Fixing some initial vertex s, let us denote by Σ^{MP} the set of all MD strategies σ satisfying

Procedure `Qual-CN`($\mathcal{D}$,r)

Data: A MDP $\mathcal{D}$ with reward r.
Result: Compute the set $A \subseteq V$ of vertices with $Val^{CN} = 1$, and a MD strategy, σ, *CN*-optimal starting at every $v \in A$.

1 $\mathcal{D}' \leftarrow$ `Decreasing`($\mathcal{D}$)
2 $(A', \sigma') \leftarrow$ `Qual-MP`($\mathcal{D}'$,r)
3 $A \leftarrow \{v \in V \mid (v, 1, 0) \in A'\}$
4 $\sigma \leftarrow$ `CN-FD-to-MD`(σ')
5 **return** (A, σ)

$\mathcal{P}(MP^\sigma(s)) = 1$, and by Σ^{CN} the set of all MD strategies σ satisfying $\mathcal{P}(CN^\sigma(s)) = 1$. It is not hard to see that $\Sigma^{CN} \subseteq \Sigma^{MP}$. If this was an equality, the reduction would boil down to the identity map. Unfortunately, these sets are not equal in general. A trivial example is provided by a MDP with just one vertex s with reward 0. More generally, the strategy σ may be trapped in a finite loop around 0 (causing $\mathcal{P}(MP^\sigma(s)) = 1$) but never accumulate all negative values (causing $\mathcal{P}(CN^\sigma(s)) = 0$). As a solution to this problem, we characterize in Lemma 3.3 the strategies from Σ^{MP} which are also in Σ^{CN}, via the property of being "decreasing":

Definition 3.3. *A MD strategy σ in $\mathcal{D}$ is* decreasing *if for every state u of $\mathcal{D}\langle\sigma\rangle$ reachable from s there is a finite path w initiated in u such that $r(w) = -1$.*

Lemma 3.3. *Σ^{CN} is the set of all decreasing strategies from Σ^{MP}.*

A key part of the reduction is the construction of an MDP, $\mathcal{D}'$, described in Figure 1, which simulates the MDP $\mathcal{D}$, but satisfies that $\Sigma^{MP} = \Sigma^{CN}$ for every initial vertex s. The idea is to augment the vertices of $\mathcal{D}$ with additional information, keeping track of whether the run under some $\sigma \in \Sigma^{MP}$ "oscillates" with accumulated rewards in a bounded neighborhood of 0, or "makes progress" towards $-\infty$. The last obstacle in the reduction is that MD strategies for $\mathcal{D}'$ do not directly yield MD strategies for $\mathcal{D}$. Rather a *CN*-optimal MD strategy, τ', for $\mathcal{D}'$ induces a deterministic *CN*-optimal strategy, τ, which uses a finite automaton to evaluate the history of play. Fortunately, given such a strategy τ it is possible to transform it to a *CN*-optimal MD strategy for $\mathcal{D}$ by carefully eliminating the memory it uses. This is done on line 4. We refer the reader to the full version of this paper [5] for proofs of this claims, and just note that the construction of $\mathcal{D}'$ on line 1, procedure `Decreasing` can clearly be done in polynomial time. Thus, the overall time complexity of the reduction is polynomial.

3.3 Solving Qualitative *MP*. For a fixed vertex $s \in V$, for every MD strategy σ and reward function r, we define a random variable $V[\sigma, r]$ such that for every run $w \in$

$\mathcal{D}' = (V', \rightsquigarrow, (V'_N, V'_P), Prob')$, where

- $V' = \{(u,n,m), [u,n,m,v] \mid u \in V, u \hookrightarrow v, 0 \le n,m \le |V|^2+1\} \cup \{div\}$
- $V'_P = \{[u,n,m,v] \in V' \mid u \in V_P\}$, $V'_N = V' \setminus V'_P$
- transition relation $\rightsquigarrow$ is the *least* set satisfying the following for every $u, v \in V$ such that $u \hookrightarrow v$ and $0 \le m,n \le |V|^2+1$:
  - if $m = |V|^2+1$ and $n > 0$, then $(u,n,m) \rightsquigarrow div$
  - if $m \le |V|^2+1$ and $n = 0$, then $(u,n,m) \rightsquigarrow [u,1,0,v]$
  - if $m < |V|^2+1$ and $n > 0$, then $(u,n,m) \rightsquigarrow [u,n,m,v]$
  - if $u \in V_P$, then $[u,n,m,v] \rightsquigarrow (v, n+r(u), m+1)$ and $[u,n,m,v'] \rightsquigarrow (v,1,0)$ for all $v' \in V \setminus \{v\}$ such that $[u,n,m,v'] \in V'$
  - if $u \in V_N$, then $[u,n,m,v] \rightsquigarrow (v, n+r(u), m+1)$
  - $div \rightsquigarrow div$

$Prob'([u,n,m,v] \rightsquigarrow (v',n',m')) = Prob(u \hookrightarrow v')$ whenever $[u,n,m,v] \in V'_P$ and $[u,n,m,v] \rightsquigarrow (v',n',m')$. Finally, $r'((u,n,m)) = 0$, $r'([u,n,m,v]) = r(u)$ and $r'(div) = 1$.

Figure 1: Definition of the MDP $\mathcal{D}'$.

$Run_{\mathcal{D}\langle\sigma\rangle}(s)$:

$$V[\sigma,r](w) = \begin{cases} \lim_{n\to\infty} \frac{r(w\downarrow n)}{n} & \text{if the limit exists;} \\ \bot & \text{otherwise.} \end{cases}$$

It follows from, e.g., [22, Theorem 1.10.2] that since σ is MD the value of $V[\sigma,r]$ is almost surely defined. Solving the *MP* objective amounts to finding a MD strategy σ such that $\mathcal{P}(V[\sigma,r] \le 0)$ is maximal among all MD strategies. We use the procedure `get-MD-min` to find for every vertex $s \in V$ and a reward function r a MD strategy ϱ such that $EV[\varrho,r] = \min_{\sigma \in MD} EV[\sigma,r]$. This can be done in polynomial time via linear programming: see, e.g., [14, Algorithm 2.9.1] or [23, Section 9.3].

The core idea of procedure `Qual-MP` for solving **MDP-MP-qual** is this: Whenever $EV[\tau,r] \le 0$ there is a bottom strongly connected component (BSCC), C, of the transition graph of $\mathcal{D}\langle\tau\rangle$, such that almost all runs w reaching C satisfy $V[\tau,r](w) \le 0$. Since $Val^{MP}(s) = 1$ implies the existence of some $\tau \in \Sigma^{MP}$ such that $EV[\tau,r] \le 0$, `Qual-MP` solves **MDP-MP-qual** by successively cutting off the BSCCs just mentioned, while maintaining the invariant $\exists \tau : EV[\tau,r] \le 0$. Detailed proofs are in the full version [5].

`Extract`(S) removes an arbitrary element of a nonempty set S and returns it, and `MD-from-edges`(T) returns an arbitrary MD strategy σ satisfying $(u,v) \in T \wedge u \in V_N \Rightarrow \sigma(u) = v$. Both these procedures can clearly be implemented in polynomial time. Thus by the earlier discussion about the complexity of `Max-Reach`, in Section 3.1, we conclude that `Qual-MP` runs in polynomial time.

Procedure `Qual-MP`$(\mathcal{D},r)$

Data: A MDP $\mathcal{D}$ with reward r.
Result: Compute the set $A \subseteq V$ of vertices with $Val^{MP} = 1$ and a MD strategy σ *MP*-optimal starting in every $v \in A$.

```
1  V? ← V, A ← ∅, T ← ∅, r̂ ← r
2  while V? ≠ ∅ do
3      s ← Extract(V?)
4      if ∃ϱ : EV[ϱ, r̂] ≤ 0 then
5          ϱ ← get-MD-min(D,r,s)
6          C ← a BSCC C of D⟨ϱ⟩ such that C ∩ A = ∅
           and P(V[ϱ, r̂] ≤ 0 | Reach_C^ϱ) = 1
7          (τ, (reach_v)_{v∈V}) ← Max-Reach(D, C ∪ A)
8          A' ← {u ∈ V | reach_u = 1}
9          for every u ∈ V_N, v ∈ V do if
           (u ∈ C ∧ v = ϱ(u)) ∨
           (u ∈ A' \ (C ∪ A) ∧ v = τ(u)) then
           T ← T ∪ {(u, v)}
10         A ← A' ∪ A
11         for every u ∈ V do if u ∈ A then r̂(u) ← 0
12         if s ∉ A then V? ← V? ∪ {s}
13 σ ← MD-from-edges(T)
14 return (A, σ)
```

4 OC-MDPs with Boundary

Fix an OC-MDP, $\mathcal{A} = (Q, \delta^{=0}, \delta^{>0}, (Q_N, Q_P), P^{=0}, P^{>0})$, and its associated MDP, $\mathcal{D}^{\rightarrow}_{\mathcal{A}}$.

Definition 4.1. (termination objectives) *The (non-selective)* termination objective*, denoted NT, consists of all runs w of $\mathcal{D}^{\rightarrow}_{\mathcal{A}}$ that eventually hit a configuration with counter value zero. Similarly, for a set $F \subseteq Q$ of* final *states we define the associated* selective termination objective*, denoted ST_F (or just ST if F is understood), consisting of all runs of $\mathcal{D}^{\rightarrow}_{\mathcal{A}}$ that hit a configuration of the form $q(0)$ where $q \in F$.*

Termination objectives are much more complicated to analyze than the *CN* objectives considered in Section 3. As mentioned in the introduction, even for purely stochastic QBDs termination probabilities can be irrational (see [10]), and we leave open *quantitative* termination problems for OC-MDPs. Even *qualitative* problems for OC-MDPs require new insights.

We define $ValOne^{NT}$ and $ValOne^{ST}$ be the sets of all $p(i) \in Q \times \mathbb{N}_0$ such that $Val^{NT}(p(i)) = 1$ and $Val^{ST}(p(i)) = 1$,

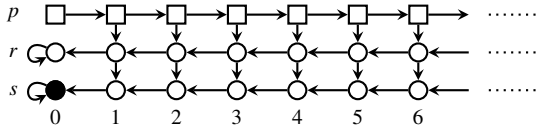
respectively. We also define their subsets $OptValOne^{NT}$ and $OptValOne^{ST}$ consisting of all $p(i) \in ValOne^{NT}$ and all $p(i) \in ValOne^{ST}$, respectively, such that there is an optimal strategy achieving value 1 (with respect to NT, and ST, respectively) starting at $p(i)$. Are the inclusions $OptValOne^{NT} \subseteq ValOne^{NT}$ and $OptValOne^{ST} \subseteq ValOne^{ST}$ proper? It turns out that the two objectives differ in this respect. We begin by stating our results about qualitative NT objectives.

THEOREM 4.1. *$ValOne^{NT} = OptValOne^{NT}$. In other words, for the non-selective termination objective, if the supremum probability of termination achievable is 1, then there is a strategy that achieves this.*

Moreover, given a OC-MDP, $\mathcal{A}$, and a configuration $q(i)$ of $\mathcal{A}$, we can decide in polynomial time whether $q(i) \in ValOne^{NT}$. Furthermore, there is a CMD strategy, σ, constructible in polynomial time, which is optimal starting at every configuration in $ValOne^{NT} = OptValOne^{NT}$.

We now sketch the proof of this theorem. Basically, when starting at a configuration $p(i)$ where the value of the counter, i, is large enough (specifically, when $i \geq |Q|$, where $|Q|$ denotes the number of states), we can show that if the supremum probability of termination starting at $p(i)$ is 1, then if we view the OC-MDP as boundaryless, the supremum probability of achieving $CN_{\mathcal{A}}$ (arbitrarily small values of the counter) starting at $p(i)$ is also 1. Furthermore, since we know we can achieve $CN_{\mathcal{A}}$ with an optimal CMD strategy, we conclude that we can achieve NT with an optimal CMD strategy, when starting at large enough counter values i. Now, observe that the set of configurations $q(j)$ such that $j \leq |Q|$ is finite (and indeed there are polynomially many such configurations). We can therefore consider these as forming a finite-state MDP, where our objective is to either reach a configuration $q(0)$ or to reach a configuration $q(|Q|)$ from which we know that we can achieve $CN_{\mathcal{A}}$ with probability 1. Since, as discussed earlier, MDP reachability problems are solvable in P-time, we are done.

Next we turn to ST objectives. First, let us observe that the inclusion $OptValOne^{ST} \subseteq ValOne^{ST}$ is proper: there may be no optimal strategy for ST even when the value is 1. Consider the OC-MDP $\hat{\mathcal{A}}$ of the following figure (we draw directly the associated MDP $\mathcal{D}^{\rightarrow}_{\hat{\mathcal{A}}}$):



The state p is controlled, and the other two states are stochastic. The probability distributions are always uniform, and the only final state is s. Now observe that $OptValOne^{ST} = \{s(i) \mid i \in \mathbb{N}_0\}$, while $ValOne^{ST}$ consists of all $p(i), s(i), i \in \mathbb{N}_0$. This is basically because starting from $p(i)$ we can first increment the counter to an arbitrarily high value $j > i$, and thus insure that once we move from $p(j)$ to $r(j)$, that we have an arbitrarily large number of chances to move (with positive probability bounded away from 0) from $r(k)$ to $s(k)$ and thereafter to $s(0)$. Thus we can make the probability of termination in $s(0)$ starting from $p(i)$ arbitrarily close to 1. But there is clearly no single strategy that achieves probability 1, because any strategy that achieves positive probability of termination has to, for some i, and with positive probability, move to $r(i)$ from $p(i)$, and thereafter with positive probability we terminate in $r(0)$ rather than $s(0)$.

We now provide an exponential time algorithm to decide whether a given configuration $q(i)$ is in $OptValOne^{ST}$, and we show that there is a "counter-regular" strategy σ constructible in exponential time that is optimal starting at all configurations in $OptValOne^{ST}$. We first introduce the notion of *coloring*.

DEFINITION 4.2. (COLORING) *A coloring is a map $C : Q \times \mathbb{N}_0 \to \{b, w, g, r\}$, where b, w, g, and r are the four different "colors" (black, white, gray, and red). For every $i \in \mathbb{N}_0$, we define the* i-th column *of C as a map $C_i : Q \to \{b, w, g, r\}$, where $C_i(q) = C(q(i))$.*

A coloring can be depicted as an infinite matrix of points (each being black, white, gray, or red) with rows indexed by control states and columns indexed by counter values. We are mainly interested in the coloring, R, which represents the set $OptValOne^{ST}$ in the sense that for every $p(i) \in Q \times \mathbb{N}_0$, the value of $R(p(i))$ is either b or w, depending on whether $p(i) \in OptValOne^{ST}$ or not. First, we show R is "ultimately periodic":

LEMMA 4.1. *Let $N = 2^{|Q|}$. There is an ℓ, $1 \leq \ell \leq N$, such that for $j \geq N$, we have $R_j = R_{j+\ell}$.*

The proof is quite simple: There are only N distinct possible colorings for each column R_j, so there must be two distinct columns such that $R_j = R_k$. Specifically, there must be $j \leq N$ and $1 \leq \ell \leq N$ such that $R_j = R_{j+\ell}$. Then we simply observe that for any k, the coloring of column R_{k+1} is entirely determined by the coloring of column R_k. This is because starting at any configuration $p(k+1)$ there is a strategy to terminate with probability 1 in the desired set of states F, if and only if there is a strategy using which we shall, with probability 1, eventually hit counter value k and furthermore the first time we do hit counter value k we shall be at a "black" configuration, i.e., one from which we have a strategy to terminate with probability 1 in the desired set F of states.

Thus the coloring R consists of an "initial rectangle" of width $N+1$ followed by infinitely many copies of the "periodic rectangle" of width ℓ (see Fig. 2). Note that $R_N = R_{N+\ell}$. Another important observation about the

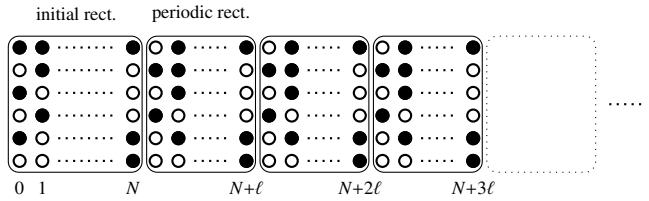


Figure 2: The structure of coloring R (where $N = 2^{|Q|}$).

coloring R is the following: let $q(m)$ be a configuration of $\mathcal{A}$ where $q \in Q_N$ and $C(q(m)) = b$. Let us call a transition $q(m) \mapsto r(n)$ *useless* if $C(r(n)) = w$. Obviously, useless transitions are never used by an optimal strategy that achieves the value 1 with respect to the considered ST objective. Let $\mathcal{D}$ be the MDP obtained from $\mathcal{D}^{\rightarrow}_{\mathcal{A}}$ by removing all useless transitions. Since the coloring R is ultimately periodic, there is a one-counter automaton $\mathcal{A}_\ell$ such that $\mathcal{D} = \mathcal{D}^{\rightarrow}_{\mathcal{A}_\ell}$. The control states of $\mathcal{A}_\ell$ are pairs (q, k) where $q \in Q$ and $0 \leq k \leq N{+}\ell$, and the zero/positive rules of $\mathcal{A}_\ell$ "encode" the structure of the initial/periodic rectangle of $\mathcal{D}^{\rightarrow}_{\mathcal{A}}$ where all useless transitions are removed. For example, $((q, N{+}\ell), 1, (r, N{+}1))$ is a positive rule of $\mathcal{A}_\ell$ iff $R(q(N{+}\ell)) = R(r(N{+}\ell{+}1)) = b$ and $q(N{+}\ell) \mapsto r(N{+}\ell{+}1)$ is a transition of $\mathcal{A}$. Hence, the counter of $\mathcal{A}_\ell$ encodes the "index" of the current periodic rectangle. One can prove that for every $q(N{+}k)$ where $1 \leq k \leq \ell$ we have that $R(q(N{+}k)) = b$ iff $Val^{CN}((q, N{+}k)(1)) = 1$, which establishes an important link to our previous results.

We show how to compute the initial and periodic rectangles of R by, intuitively, trying out all (exponentially many) candidates for the width ℓ and the columns $R_N = R_{N+\ell}$. For each such pair of candidates, the algorithm tries to determine the color of the remaining points in the initial and periodic rectangles, until it either finds an inconsistency with the current candidates, or produces a coloring which is not necessarily the same as R, but where all black points are certified by an optimal strategy. Since the algorithm eventually tries also the "real" ℓ and $R_N = R_{N+\ell}$, all black points of R are discovered.

Let us briefly discuss the way colors of points in the periodic rectangle are determined. First, the color of all these points is initialized to gray, which represents the "don't know" value. Then, the algorithm tries to recolor as many gray points as possible to white or black by looking at the current color of the immediate successors of a given point. For example, a gray stochastic vertex is recolored to black/white if all/some of its successors are black/white. Note that this procedure can also fail in the sense that the same point may need to be recolored both to black and white, in which case the algorithm immediately reports an inconsistency in the current choice of candidates for the ℓ and the columns $R_N = R_{N+\ell}$. (The red color is used for flagging such inconsistencies. We will not indicate here how this is done in any more detail.) Otherwise, the periodic rectangle is recolored so that all points are either black, white, or gray, and the color of each point is consistent with the colors of its immediate successors (for example, each controlled point which is black or gray can only have successors that are black or gray), and no point can be further recolored in the above described way. Now, the algorithm constructs the one-counter automaton $\mathcal{A}_\ell$ discussed above (using the current candidate for ℓ and treating all gray points as if they were black). For every black or gray point $q(N{+}k)$ we now compute the associated CN-value $Val^{CN}((q, N{+}k)(1))$. Note that this can be done in time polynomial in the size of $\mathcal{A}_\ell$ and hence exponential in the size of $\mathcal{A}$ by applying the results of Section 3. If we discover a black point such that the associated CN-value is not 1, the algorithm reports an inconsistency in the current choice of candidates for the ℓ and the columns $R_N = R_{N+\ell}$. Otherwise, each gray point is recolored to black or white depending on whether its associated CN-value is equal to 1 or not, respectively.

Similarly, we determine the color of the remaining points in the initial rectangle. A full description of the algorithm, along with a discussion of its many subtleties and a proof of correctness, is given in the full version [5].

THEOREM 4.2. *An automaton recognizing $OptValOne^{ST}$, and a counter-regular strategy, σ, optimal starting at every configuration in $OptValOne^{ST}$, are both computable in exponential time.*

Thus, membership in $OptValOne^{ST}$ is solvable in exponential time. We do not have an analogous result for $ValOne^{ST}$ and leave this as an open problem (the example earlier which showed that in general $ValOne^{ST} \neq OptValOne^{ST}$, gives a taste of the difficulties).

A straightforward reduction from the emptiness problem for alternating finite automata over a one-letter alphabet, which is **PSPACE**-hard, see e.g. [17], shows that membership in $OptValOne^{ST}$ is **PSPACE**-hard. Further, we show that membership in $ValOne^{ST}$ is hard for the Boolean Hierarchy (**BH**) over **NP**, and thus neither in **NP** nor **coNP** assuming standard complexity assumptions. The proof technique, based on a number-theoretic encoding, originated in [18] and was used in [16, 24].

THEOREM 4.3. *Membership in $ValOne^{ST}$ is **BH**-hard. Membership in $OptValOne^{ST}$ is **PSPACE**-hard.*

For the special subclass of OC-MDPs consisting of *solvency games* [1], we show all *qualitative* problems are decidable in polynomial time. A *solvency game*, is given by a positive integer, n, (the initial wealth of the "investor" or "gambler"), and a finite set $\mathcal{A} = \{A_1, \ldots, A_k\}$ of *actions* (or "gambles"), each of which is associated with a finite-support probability distribution on the integers. (So, we can

equate an action A_i with the random variable that samples from its associated distribution.) We assume the distribution associated with each action A_i, is encoded by giving a set of pairs $\{(n_{i,1}, p_{i,1}), (n_{i,2}, p_{i,2}), \ldots, (n_{i,m_i}, p_{i,m_i})\}$, such that for $j = 1, \ldots, m_i$, $n_{i,j} \in \mathbb{Z}$ and $p_{i,j}$ are positive rational probabilities, and $\sum_{j=1}^{m_i} p_{i,j} = 1$. We assume integers and rational probabilities are encoded in the standard way in binary. The investor with wealth n has to repeatedly choose an action (gamble) from the set $\mathcal{A}$. If at any time the current wealth is $n' > 0$ and the gambler chooses an action A_i, then we sample from the distribution of A_i and the resulting integer is added to n' to determine the new wealth. If wealth ever hits 0 or becomes negative, play immediately stops: the investor is bankrupt. The investor's objective is to minimize the probability of bankruptcy. As discussed in the introduction, it is easy to see that solvency games form a subclass of OC-MDPs.

PROPOSITION 4.1. *Given a solvency game, it is (easily) decidable in polynomial time whether the gambler has a strategy to not go bankrupt with probability:* > 0, $= 1$, $= 0$, *or* < 1.

The cases other than > 0 are either trivial or follow very easily from what we have established for OC-MDPs. Indeed, there is a strategy to not go bankrupt with probability $= 1$ (< 1, respectively) iff there is an action A_i that does not have (that has, respectively) a negative integer in its support. Also, there is a strategy to not go bankrupt with probability $= 0$, i.e., to almost surely go bankrupt, iff there is an action A_i that has a negative integer in its support and furthermore its expectation (or *drift*) is nonpositive, i.e., $E[A_i] \leq 0$. (This is because, by our results for OC-MDPs, a CMD optimal strategy suffices for maximizing termination probability, and this implies that for the special case of solvency games repeating some particular action A_i forever is an optimal strategy for maximizing the probability of bankruptcy.)

For the more interesting case, > 0, we make use of a lovely theorem on *controlled* random walks by Durrett, Kesten, and Lawler [9, Theorem 1]. Their theorem says that if we can choose an infinite sequence of independent random variables X_i, $i = 1, \ldots$ each from a finite set of distributions $F_1, \ldots, F_k$, each having expectation 0 and bounded variance, then even if we can make our choices *adaptively* based on outcomes of earlier choices, the sums $S_n = \sum_{i=1}^{n} X_i$ will have the property that they are *recurrent*, meaning that values close to 0 recur infinitely often with probability 1. The statement of their theorem looks intuitively obvious, but their proof is quite non-trivial. (It makes use of Skorokhod's theorem on the embeddability of any 0 expectation and bounded-variance random variable inside Brownian motion. It is worth noting that when variance is not bounded their theorem can fail, and they give examples of this.) Using their theorem, we can conclude that there is a strategy to not go bankrupt with probability > 0 in a solvency game iff there is either a trivial action $A_i \equiv 0$, using which the wealth stays unchanged with probability 1, or there is some action A_i available whose expected drift satisfies $E[A_i] > 0$. Repeating that action forever will yield positive probability of not going bankrupt, and since actions have finite support and thus bounded variance, [9, Theorem 1] easily implies one of these two conditions is also necessary.

Acknowledgements. We thank Petr Jančar, Richard Mayr, and Olivier Serre for pointing out the PSPACE-hardness of $OptValOne^{ST}$. Václav Brožek acknowledges support by a Newton International Fellowship from the Royal Society.

References

[1] N. Berger, N. Kapur, L. J. Schulman, and V. Vazirani. Solvency Games. In *Proc. of FSTTCS'08*, 2008.

[2] P. Billingsley. *Probability and Measure*. J. Wiley and Sons, 3rd edition, 1995.

[3] D. Bini, G. Latouche, and B. Meini. *Numerical methods for Structured Markov Chains*. Oxford University Press, 2005.

[4] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. Williamson. Adversarial queuing theory. *J. ACM*, 48(1):13–38, 2001.

[5] Tomáš Brázdil, Václav Brožek, Kousha Etessami, Antonín Kučera, and Dominik Wojtczak. One-Counter Markov Decision Processes. *CoRR*, abs/0904.2511, 2009. http://arxiv.org/abs/0904.2511.

[6] T. Brázdil, V. Brožek, V. Forejt, and A. Kučera. Reachability in recursive Markov decision processes. In *Proc. 17th Int. CONCUR*, pages 358–374, 2006.

[7] K. L. Chung. *A Course in Probability Theory*. Academic Press, 3rd edition, 2001.

[8] C. Courcoubetis and M. Yannakakis. Markov decision processes and regular events. *IEEE Trans. on Automatic Control*, 43(10):1399–1418, 1998.

[9] R. Durrett, H. Kesten, and G. Lawler. Making money from fair games. In *Random Walks, Brownian Motion, and Interacting Particle Systems*, pages 255-267, *Progress in Probability* vol. 28, R. Durrett and H. Kesten, editors, Birkhäuser, 1991.

[10] K. Etessami, D. Wojtczak, and M. Yannakakis. Quasi-birth-death processes, tree-like QBDs, probabilistic 1-counter automata, and pushdown systems. In *Proc. 5th Int. Symp. on Quantitative Evaluation of Systems (QEST)*, pages 243–253, 2008.

[11] K. Etessami and M. Yannakakis. Recursive Markov decision processes and recursive stochastic games. In *Proc. 32nd Int. Coll. on Automata, Languages, and Programming (ICALP)*, pages 891–903, 2005.

[12] K. Etessami and M. Yannakakis. Efficient qualitative analysis of classes of recursive Markov decision processes and simple stochastic games. In *Proc. of 23rd STACS'06*. Springer, 2006.

[13] W. Feller. *An Introduction to Probability Theory and its Applications*, volume 1. Wiley & Sons, 1968.

[14] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1997.

[15] H. Gimbert. Pure stationary optimal strategies in markov decision processes. In *STACS*, pages 200–211, 2007.
[16] P. Jančar, A. Kučera, F. Moller, and Z. Sawa. DP lower bounds for equivalence-checking and model-checking of one-counter automata. *Inf. Comput.*, 188(1):1–19, 2004.
[17] P. Jančar, and Z. Sawa. A note on emptiness for alternating finite automata with a one-letter alphabet *Information Processing Letters* 104(5):164–167, Elsevier, 2007.
[18] A. Kučera. The complexity of bisimilarity checking for one-counter processes. *Theo. Comp. Sci.*, 304:157–183, 2003.
[19] J. Lambert, B. Van Houdt, and C. Blondia. A policy iteration algorithm for markov decision processes skip-free in one direction. In *Numerical Methods for Structured Markov Chains*, 2007.
[20] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM series on statistics and applied probability, 1999.
[21] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models:an algorithmic approach*. Johns Hopkins U. Press, 1981.
[22] J. R. Norris. *Markov chains*. Cambridge University Press, 1998.
[23] M. L. Puterman. *Markov Decision Processes*. Wiley, 1994.
[24] O. Serre. Parity games played on transition graphs of one-counter processes. In *FoSSaCS*, pages 337–351, 2006.
[25] V. Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge U. Press, 2nd edition, 2008.
[26] L. G. Valiant and M. Paterson. Deterministic one-counter automata. In *Automatentheorie und Formale Sprachen*, volume 2 of *LNCS*, pages 104–115. Springer, 1973.
[27] L. B. White. A new policy iteration algorithm for Markov decision processes with quasi birth-death structure. *Stochastic Models*, 21:785–797, 2005.