

One-Counter Markov Decision Processes

T. Brázdil¹, V. Brožek¹, K. Etessami², A. Kučera¹, and D. Wojtczak^{2,3}

¹ Faculty of Informatics, Masaryk University
{xbrazdil,xbrozek,tony}@fi.muni.cz

² School of Informatics, University of Edinburgh
kousha@inf.ed.ac.uk

³ CWI, Amsterdam
D.K.Wojtczak@cwi.nl

Abstract. We study the computational complexity of 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 \limsup 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.⁴ 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, 15]). 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 [9] for 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

⁴ 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.

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 in fact radically alter the behavior of the model, even for purely probabilistic QBDs (see appendix C 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 Section 4, Proposition 17, 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 is 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 [9] 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 [9] 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 \limsup value is $= \infty$ (or, by symmetry, that the \liminf 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 ([14]), 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 \limsup (\liminf) 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 [9].
 - (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 not 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) The optimal probabilities can of course again be irrational.
 - (b) 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).
 - (c) Even deciding whether there is an optimal strategy which ensures probability 1 termination in the selected states is PSPACE-hard.
 - (d) 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 fascinating 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 $> p$. 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 likely 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 [10, 11]. 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 [10]. 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), [10] showed quantitative problems for optimal termination probability are decidable in PSPACE, and [11] showed that deciding whether the optimal termination probability is 1 can be done in P-time. In [5] this was extended further to answer 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). 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 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 $len(w)$, where the length of an infinite word is ∞ . Given a word (finite or infinite) over Σ , the individual letters of w are denoted $w(0), w(1), \dots$ (so indexing begins at 0). For a word w , we denote by $w \downarrow n$ the prefix $w(0) \dots w(n-1)$ of w . Let $\mathcal{V} = (V, \rightarrow)$ where V is a non-empty set and $\rightarrow \subseteq V \times V$ a *total* relation (i.e., for every $v \in V$ there is some $u \in V$ such that $v \rightarrow u$). The reflexive transitive closure of \rightarrow is denoted \rightarrow^* . A *path* in \mathcal{V} is a finite or infinite word $w \in V^+ \cup V^\omega$ such that $w(i-1) \rightarrow w(i)$ for every $1 \leq i < len(w)$. 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 $f : X \rightarrow [0, 1]$ 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 $\mathcal{M} = (S, \rightarrow, Prob)$ where S is a finite or countably infinite set of *states*, $\rightarrow \subseteq S \times S$ 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 $s \rightarrow t$ and x is the probability of $s \rightarrow t$. 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 $\mathcal{P} : \mathcal{F} \rightarrow [0, 1]$ is the unique probability measure such that $\mathcal{P}(Run_{\mathcal{M}}(w)) = \prod_{i=1}^{len(w)-1} x_i$ where $w(i-1) \xrightarrow{x_i} w(i)$ for every $1 \leq i < len(w)$. If $len(w) = 1$, we put $\mathcal{P}(Run_{\mathcal{M}}(w)) = 1$.

Definition 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 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, and we use similar suggestive notation for other strategy classes.

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 $Prob(u, u') = x$, or (2) $u \in V_N$ and $\sigma(wu)$ assigns x to the transition (u, 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*⁵ 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. 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 $\mathcal{D}_{\mathcal{A}}^{\rightarrow}$ and $\mathcal{D}_{\mathcal{A}}^{\leftrightarrow}$ as follows:

- $\mathcal{D}_{\mathcal{A}}^{\rightarrow} = (Q \times \mathbb{N}_0, \mapsto, (Q_N \times \mathbb{N}_0, Q_P \times \mathbb{N}_0), Prob)$. 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 $P^{>0}(p, j-i, q)$.
- $\mathcal{D}_{\mathcal{A}}^{\leftrightarrow} = (Q \times \mathbb{Z}, \mapsto, (Q_N \times \mathbb{Z}, Q_P \times \mathbb{Z}), Prob)$, 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 $\mathcal{D}_{\mathcal{A}}^{\rightarrow}$ and $\mathcal{D}_{\mathcal{A}}^{\leftrightarrow}$ 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 $\mathcal{D}_{\mathcal{A}}^{\rightarrow}$ or $\mathcal{D}_{\mathcal{A}}^{\leftrightarrow}$, *counter-oblivious-MD* (denoted *CMD*) if there is a *selector*, $f : Q \rightarrow \delta^{>0}$ (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 prob. 1 (ignoring history and n).

⁵ 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 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 \liminf counter value is $= -\infty$). Our goal is to prove Theorem 4. (All proofs missing in this section can be found in the Appendix.)

Definition 3. Let \mathcal{A} be a OC-MDP. We use $CN_{\mathcal{A}}$ to denote the set of all runs $w \in \text{Run}_{\mathcal{D}^{\leftrightarrow \mathcal{A}}}$ 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 4. Given a OC-MDP, \mathcal{A} , there is a $CN_{\mathcal{A}}$ -optimal CMD strategy for it, which is computable in polynomial time. Moreover, $\text{Val}^{CN_{\mathcal{A}}}$ 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 an MDP with reward is equipped with $r : V \rightarrow \{-1, 0, 1\}$. For $v = v_0 \cdots v_n \in V^+$, let $r(v) := \sum_{i=0}^n r(v_i)$.

Definition 5. We denote by CN the set of all $w \in \text{Run}_{\mathcal{D}}$ satisfying $\liminf_{n \rightarrow \infty} r(w \downarrow n) = -\infty$. We further denote by MP the set of all runs $w \in \text{Run}_{\mathcal{D}}$ such that $\lim_{n \rightarrow \infty} \frac{r(w \downarrow n)}{n}$ exists and $\lim_{n \rightarrow \infty} \frac{r(w \downarrow n)}{n} \leq 0$.⁶

A theorem by Gimbert ([14, Theorem 1]) implies there is always a CN -optimal MD strategy for finite MDPs, because (the characteristic function of) objective CN is *prefix-independent* and *submixing* (see Section A.2). Lemma 7 shows for OC-MDPs there is always a $CN_{\mathcal{A}}$ -optimal CMD strategy. We define several problems:

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 $\text{Val}^{CN_{\mathcal{A}}}(p(z))$, 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 $\text{Val}^{CN}(v)$, for every vertex v of \mathcal{D} .

MDP-CN-qual:

Input: finite-state MDP, \mathcal{D} , with reward function r .

Output: set $A = \{v \mid \text{Val}^{CN}(v) = 1\}$, 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 $\forall v \in A : \mathcal{P}(MP^{\bar{\sigma}}(v)) = 1$.⁷

Proposition 6. 1. There exist the following polynomial-time (Turing) reductions:

$$OC\text{-MDP-CN} \leq_P MDP\text{-CN} \leq_P MDP\text{-CN-qual} \leq_P MDP\text{-MP-qual}$$

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

The following lemma establishes both the first reduction of Proposition 6, part 1, and the existence of $CN_{\mathcal{A}}$ -optimal CMD strategies for OC-MDPs.

Lemma 7. 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 $\text{Val}^{CN_{\mathcal{A}}}(p(i)) = \text{Val}^{CN}(p)$. 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 $p(i) \in Q \times \mathbb{Z}$, $\mathcal{P}(CN_{\mathcal{A}}^{\sigma'}(p(i))) = \mathcal{P}(CN^{\sigma}(p))$.

⁶ ‘‘MP’’ stands for ‘‘(non-positive) Mean Payoff’’.

⁷ The existence of strategy $\bar{\sigma}$ is a consequence of the correctness proof in Section A.7.

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

Data: A MDP \mathcal{D} with reward r .

Result: Compute the vector $(Val^{CN}(v))_{v \in V}$, and a CN-optimal MD strategy, σ .

- 1 $(A, \tau) \leftarrow \text{Qual-CN}(\mathcal{D}, r)$
 - 2 $(\sigma_R, (val_v)_{v \in V}) \leftarrow \text{Max-Reach}(\mathcal{D}, A)$
 - 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** $(val_v)_{v \in V}, \sigma$
-

Dealing with MD strategies simplifies notation. Although the Markov chain $\mathcal{D}(\sigma)$ has infinitely many states, for a finite MDP $\mathcal{D} = (V, \hookrightarrow, (V_N, V_P), Prob)$ and a MD strategy σ we can replace $\mathcal{D}(\sigma)$ with a finite-state Markov chain $\mathcal{D}(\sigma)$ where V is the set of states, and $u \xrightarrow{\sigma} u'$ iff $u \xrightarrow{\sigma} uu'$ 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 $Run_{\mathcal{D}(\sigma)}(u)$ given by the bijection of runs which maps run w to $w_{\mathcal{D}}$, see the definition of $\mathcal{D}(\sigma)$ in Sect. 2.

To finish the proof of Theorem 4 we have to provide the last two reductions from Proposition 6, part 1, prove that Val^{CN} is always rational, and prove Proposition 6, part 2. We do these in separate subsections.

3.1 Reduction to Qualitative CN

Proposition 8. *Let $A := \{v \in V \mid Val^{CN}(v) = 1\}$. Then for all $u \in V$ we have:*

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

The reduction **MDP-CN** \leq_P **MDP-CN-qual** is described in procedure **Solve-CN**. Its correctness follows from Proposition 8. 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: both an optimal strategy and the associated optimal (rational) probabilities can be obtained by solving suitable linear programs (see, e.g., [7] 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 7 implies that $Val^{CN_{\mathcal{R}}}$ 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 $\mathcal{P}(MP^\sigma(s)) = 1$, 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 10 the strategies from Σ^{MP} which are also in Σ^{CN} , via the property of being “decreasing”:

Definition 9. *A MD strategy σ in \mathcal{D} is decreasing if for every state u of $\mathcal{D}(\sigma)$ reachable from s there is a finite path w initiated in u such that $r(w) = -1$.*

Lemma 10. *Σ^{CN} is the set of all decreasing strategies from Σ^{MP} .*

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

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

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

Fig. 1. Definition of the MDP \mathcal{D}' .

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 postpone the proof of these claims to the Appendix, 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 $w \in \text{Run}_{\mathcal{D}(\sigma)}(s)$:

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

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 $\mathcal{P}(V[\sigma, r] \leq 0)$ 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., [13, Algorithm 2.9.1] or [23, Section 9.3].

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  $V_? \leftarrow V, A \leftarrow \emptyset, T \leftarrow \emptyset, \hat{r} \leftarrow r$ 
2 while  $V_? \neq \emptyset$  do
3    $s \leftarrow \text{Extract}(V_?)$ 
4   if  $\exists \varrho : EV[\varrho, \hat{r}] \leq 0$  then
5      $\varrho \leftarrow \text{get-MD-min}(\mathcal{D}, r, s)$ 
6      $C \leftarrow$  a BSCC  $C$  of  $\mathcal{D}(\varrho)$  such that  $C \cap A = \emptyset$  and  $\mathcal{P}(V[\varrho, \hat{r}] \leq 0 \mid \text{Reach}_C^e) = 1$ 
7      $(\tau, (\text{reach}_v)_{v \in V}) \leftarrow \text{Max-Reach}(\mathcal{D}, C \cup A)$ 
8      $A' \leftarrow \{u \in V \mid \text{reach}_u = 1\}$ 
9     for every  $u \in V_N, v \in V$  do if  $(u \in C \wedge v = \varrho(u)) \vee (u \in A' \setminus (C \cup A) \wedge v = \tau(u))$  then  $T \leftarrow T \cup \{(u, v)\}$ 
10     $A \leftarrow A' \cup A$ 
11    for every  $u \in V$  do if  $u \in A$  then  $\hat{r}(u) \leftarrow 0$ 
12    if  $s \notin A$  then  $V_? \leftarrow V_? \cup \{s\}$ 
13  $\sigma \leftarrow \text{MD-from-edges}(T)$ 
14 return  $(A, \sigma)$ 
```

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

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.

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, $\mathcal{D}_{\mathcal{A}}^{\rightarrow}$.

Definition 11 (termination objectives). *The (non-selective) termination objective, denoted NT , consists of all runs w of $\mathcal{D}_{\mathcal{A}}^{\rightarrow}$ 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 $\mathcal{D}_{\mathcal{A}}^{\rightarrow}$ that hit a configuration of the form $q(0)$ where $q \in F$.*

Termination objectives are more complicated than the CN objectives considered in Section 3, and even qualitative problems for them 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 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 12. $ValOne^{NT} = OptValOne^{NT}$. 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 $ValOne^{NT} = OptValOne^{NT}$.

Next we turn to ST objectives. First, the inclusion $OptValOne^{ST} \subseteq ValOne^{ST}$ is proper: there may be no optimal strategy for ST even when the value is 1. See Appendix B for an example that establishes this. We provide an exponential time algorithm to decide whether a given configuration $q(i)$ is in $OptValOne^{ST}$, and we show 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 13 (coloring). A coloring is a map $C : Q \times \mathbb{N}_0 \rightarrow \{b, w, g, r\}$, 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 $C_i : Q \rightarrow \{b, w, g, r\}$, 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 14. Let $N = 2^{|Q|}$. There is an $\ell, 1 \leq \ell \leq N$, such that for $j \geq N$, we have $R_j = R_{j+\ell}$.

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 in appendix B). Note that $R_N = R_{N+\ell}$. 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. We note that the polynomial-time algorithm for CN objectives is used as a “black-box” here and applied to various OC-MDPs constructed from \mathcal{A} and the current coloring maintained by the algorithm (see Fig. 3). The many subtleties are discussed in Appendix B.

Theorem 15. An automaton recognizing $OptValOne^{ST}$, and a counter-regular strategy, σ , optimal starting at very 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 in appendix B 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 16. Membership in $ValOne^{ST}$ is **BH**-hard. Membership in $OptValOne^{ST}$ is **PSPACE**-hard.

As noted in the introduction, for the very special subclass of *solvency games* [1], all *qualitative* problems are decidable in polynomial time (see Appendix B for formal definitions and proofs):

Proposition 17. Given a solvency game, it is decidable in polynomial time whether the gambler has a strategy to go bankrupt with probability: > 0 , $= 1$, $= 0$, or < 1 .

The cases other than < 1 are either trivial or follow easily from what we have established for OC-MDPs. For the case < 1 , we make use of a lovely theorem on inhomogeneous (controlled) random walks [8].

Acknowledgement. The authors thank Petr Jančar, Richard Mayr, and Olivier Serre for pointing out the PSPACE-hardness of the membership problem for $OptValOne^{ST}$.

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. T. Brázdil, V. Brožek, V. Forejt, and A. Kučera. Reachability in recursive Markov decision processes. In *Proc. 17th Int. CONCUR*, pages 358–374, 2006.
6. K. L. Chung. *A Course in Probability Theory*. Academic Press, 3rd edition, 2001.
7. C. Courcoubetis and M. Yannakakis. Markov decision processes and regular events. *IEEE Trans. on Automatic Control*, 43(10):1399–1418, 1998.
8. 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.
9. 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.
10. 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.
11. 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.
12. W. Feller. *An Introduction to Probability Theory and its Applications*, volume 1. Wiley & Sons, 1968.
13. J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1997.
14. H. Gimbert. Pure stationary optimal strategies in markov decision processes. In *STACS*, pages 200–211, 2007.
15. D. Gross and C. M. Harris. *Fundamentals of Queueing Theory*. John Wiley & Sons, 3rd edition, 1998.
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.

A Proofs of Section 3

A.1 Proof of Lemma 7

Lemma 7. *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 $Val^{CN, \mathcal{A}}(p(i)) = Val^{CN}(p)$. Moreover, for a MD strategy σ in \mathcal{D} , let σ' be the CMD strategy in $\mathcal{D}_{\mathcal{A}}^{\leftrightarrow}$ with a selector f defined by $f(p) = \sigma(p)$. Then for each $p(i) \in Q \times \mathbb{Z}$, $\mathcal{P}(CN^{\sigma'}(p(i))) = \mathcal{P}(CN^{\sigma}(p))$.*

Proof. Consider a MDP $\mathcal{D} = (Q \cup \delta^{>0}, \hookrightarrow, (Q_N \cup \delta^{>0}, Q_P), Prob)$ where

$$\hookrightarrow := \{(p, (p, d, q)) \mid (p, d, q) \in \delta^{>0}\} \cup \{((p, d, q), q) \mid (p, d, q) \in \delta^{>0}\}$$

and $Prob(p, (p, d, q)) = P^{>0}(p, d, q)$ for every $p \in Q_P$. Consider a reward function $r : (Q \cup \delta^{>0}) \rightarrow \{-1, 0, 1\}$ such that $r(p) = 0$ for $p \in Q$, and $r((p, d, q)) = d$ for $(p, d, q) \in \delta^{>0}$.

Consider $\mathcal{D}_{\mathcal{A}}^{\leftrightarrow} = (Q \times \mathbb{Z}, \mapsto, (Q_N \times \mathbb{Z}, Q_P \times \mathbb{Z}), Prob)$. Let Θ be a mapping of paths in \mathcal{D} to paths in $\mathcal{D}_{\mathcal{A}}^{\leftrightarrow}$ defined as follows: Given a finite path $\omega = p_1(p_1, d_1, p_2)p_2(p_2, d_2, p_3) \cdots (p_{n-1}, d_{n-1}, p_n)p_n$ in \mathcal{D} , we define $\Theta(\omega)$ to be the path $p_1(i)p_2(i + d_1) \cdots p_n(i + \sum_{j=1}^{n-1} d_j)$. Observe that the mapping is one-to-one and onto.

Let $\bar{\sigma}$ be a HR strategy in $\mathcal{D}_{\mathcal{A}}^{\leftrightarrow}$. We define a strategy σ in \mathcal{D} as follows: For every path $\omega = p_1(p_1, d_1, p_2)p_2(p_2, d_2, p_3) \cdots (p_{n-1}, d_{n-1}, p_n)p_n$ in \mathcal{D} we have that $\sigma(\omega)$ assigns x to a transition $(p_n, (p_n, d, q))$ iff $\bar{\sigma}(\Theta(\omega))$ assigns x to $(p_n(i + \sum_{j=1}^{n-1} d_j), q(i + \sum_{j=1}^{n-1} d_j + d))$. Let us extend Θ to runs $w \in Run_{\mathcal{D}(\sigma)}(p)$ by $\Theta(w)(i) = \Theta(w(2i))$. Then $\Theta : Run_{\mathcal{D}(\sigma)}(p) \rightarrow Run_{\mathcal{D}_{\mathcal{A}}^{\leftrightarrow}(\bar{\sigma})}(p(i))$ is a bijection and induces an isomorphism of the corresponding probability spaces.⁸ Also, $\Theta(CN^{\sigma}(p)) = CN^{\bar{\sigma}}_{\mathcal{A}}(p(i))$. Thus $\mathcal{P}(CN^{\sigma}(p)) = \mathcal{P}(CN^{\bar{\sigma}}_{\mathcal{A}}(p(i)))$, and hence $Val^{CN}(p) \geq Val^{CN, \mathcal{A}}(p(i))$ because $\bar{\sigma}$ was arbitrary.

Let σ be a HR strategy in \mathcal{D} . We define a strategy $\bar{\sigma}$ in $\mathcal{D}_{\mathcal{A}}^{\leftrightarrow}$ as follows: For every path $\omega' = p_1(i)p_2(i + d_1) \cdots p_n(i + \sum_{j=1}^{n-1} d_j)$ in $\mathcal{D}_{\mathcal{A}}^{\leftrightarrow}$ we have that $\bar{\sigma}(\omega')$ assigns x to $(p_n(i + \sum_{j=1}^{n-1} d_j), q(i + \sum_{j=1}^{n-1} d_j + d))$ iff $\sigma(\Theta^{-1}(\omega'))$ assigns x to $(p_n, (p_n, d, q))$. Similarly as above, $\mathcal{P}(CN^{\sigma}(p)) = \mathcal{P}(\Theta(CN^{\sigma}(p))) = \mathcal{P}(CN^{\bar{\sigma}}_{\mathcal{A}}(p(i)))$. It follows that $Val^{CN}(p) \leq Val^{CN, \mathcal{A}}(p(i))$ because σ was arbitrary. This finishes the proof of 1.

For 2., note that if σ is a MD strategy, then the strategy $\bar{\sigma}$ defined in the previous paragraph coincides with the strategy σ' from the statement of the lemma on paths of $\mathcal{D}_{\mathcal{A}}^{\leftrightarrow}$. However, then $\mathcal{P}(CN^{\sigma}(p)) = \mathcal{P}(CN^{\bar{\sigma}}_{\mathcal{A}}(p(i))) = \mathcal{P}(CN^{\sigma'}_{\mathcal{A}}(p(i)))$.

A.2 Proof of existence of CN-optimal MD strategies

We prove that the existence of a CN-optimal MD strategy for finite-state MDPs with rewards follows from [14, Theorem 1]. To do so we need to introduce the following notions from [14]. Note that the notions are simplified to achieve an easier formulation but all the arguments can be easily modified to use the original notions.

Let $O \subseteq Run_{\mathcal{D}}$ be a measurable objective. We say that O is *positional* if there is some MD strategy $\bar{\sigma}$ such that every $v \in V$ satisfies $\mathcal{P}(O^{\bar{\sigma}}(v)) = \sup_{\sigma \in HR} \mathcal{P}(O^{\sigma}(v))$. Moreover O is *prefix independent* if for every run $w \in Run_{\mathcal{D}}$ and every finite path w' such that $w'w$ is a run we have that $w \in O$ iff $w'w \in O$. Finally, O is *submixing* if for every infinite sequence of finite paths $u_0, v_0, u_1, v_1, \dots$ such that $u_0v_0u_1v_1 \cdots$, $u_0u_1 \cdots$ and $v_0v_1 \cdots$ are runs the following is true: If $u_0v_0u_1v_1 \cdots \in O$, then $u_0u_1 \cdots \in O$, or $v_0v_1 \cdots \in O$. Theorem 1 of [14] implies that every prefix independent submixing objective is positional⁹.

⁸ I.e. for any $A \subseteq Run_{\mathcal{D}(\sigma)}(p)$ we have that A is measurable iff $\Theta(A)$ is measurable and $\mathcal{P}(A) = \mathcal{P}(\Theta(A))$.

⁹ Note that the results of [14] are more general and consider measurable pay-off functions on runs instead of sets of runs. However, if O is prefix-independent and submixing according to the definition given here, then clearly the characteristic function of O is a prefix independent and submixing pay-off function, as defined in [14], and hence the results of [14] apply.

CN is clearly prefix independent. We now prove that it is also submixing. Let $w = u_0v_0u_1v_1 \dots$ be a run. For $n \in \mathbb{N}$ we denote $u \downarrow n$ the subword of $w \downarrow n$ obtained by leaving out all v_i -parts. Similarly we denote $v \downarrow n$ the subword of $w \downarrow n$ obtained by leaving out all u_i -parts. Note that $r(w \downarrow n) = r(u \downarrow n) + r(v \downarrow n)$. However, then clearly either $\liminf_{n \rightarrow \infty} r(u \downarrow n) = -\infty$, or $\liminf_{n \rightarrow \infty} r(v \downarrow n) = -\infty$. It follows that either $u_0u_1 \dots \in CN$, or $v_0v_1 \dots \in CN$, i.e., CN is submixing. We therefore have:

Lemma 18 (cf. [14]). *For finite-state MDPs with rewards, there always exists a CN -optimal MD strategy.*

A.3 Auxiliary lemma concerning CN objectives and MD strategies

Lemma 19. *Let σ be a MD strategy in \mathcal{D} and let C be a bottom strongly connected component (BSCC) of $\mathcal{D}(\sigma)$. Given $u \in C$, we define $R_u^\sigma : \text{Run}_{\mathcal{D}(\sigma)}(u) \rightarrow \mathbb{R}$ to be a random variable giving the reward accumulated before the run returns to u , i.e.,*

$$R_u^\sigma(w) = \begin{cases} r(w \downarrow n) & \text{if } n = \min\{j \geq 1 \mid w(j) = u\} < \infty \\ \infty & \text{otherwise} \end{cases}$$

Then there is $x_C \in \{0, 1\}$ such that for all $u \in C$ we have $\mathcal{P}(CN^\sigma(u)) = x_C$. Moreover, $x_C = 1$ iff for some $u \in C$ we have $\mathcal{P}(R_u^\sigma < 0) > 0$ and $ER_u^\sigma \leq 0$ (here ER_u^σ is the expected value of R_u^σ).

Proof. Let us fix $u \in C$. From [22, Theorem 1.10.2] we have that $\mathcal{P}(\text{Reach}_{\{u\}}^\sigma(v)) = 1$ for all $v \in C$. Thus we have $\mathcal{P}(CN^\sigma(u)) = \mathcal{P}(CN^\sigma(v))$ because CN is prefix independent, moreover $\mathcal{P}(R_u^\sigma = \infty) = 0$. Hence, it suffices to show that $\mathcal{P}(CN^\sigma(u)) \in \{0, 1\}$, and that $\mathcal{P}(CN^\sigma(u)) = 1$ iff $\mathcal{P}(R_u^\sigma < 0) > 0$ and $ER_u^\sigma \leq 0$.

We define sequences of random variables $I_1, I_2, I_3 \dots$ and X_1, X_2, \dots as follows: given a run $w \in \text{Run}_{\mathcal{D}(\sigma)}(u)$, we define $I_1(w) = 0$, and for all $n \geq 2$ we define $I_n(w)$ to be the least $m > I_{n-1}(w)$ such that $w(m) = u$. We define $X_n(w) = r(w \downarrow I_{n+1}(w)) - r(w \downarrow I_n(w))$ the reward accumulated between the n -th visit to u (inclusive) and $n+1$ -th visit to u (non-inclusive). Observe that $X_1 = R_u^\sigma$ and that the variables X_1, X_2, \dots are identically distributed and independent. Therefore, the sequence X_1, X_2, \dots determines a random walk S_0, S_1, S_2, \dots on \mathbb{Z} where $S_n = \sum_{i=1}^n X_i$.

Suppose that $\mathcal{P}(R_u^\sigma < 0) > 0$ and $ER_u^\sigma \leq 0$. There are two cases depending on whether $\mathcal{P}(R_u^\sigma > 0) = 0$, or not. First, assume that $\mathcal{P}(R_u^\sigma > 0) = 0$ and thus also $EX_1 = EX_j < 0$ for all j . Then almost all $w \in \text{Run}_{\mathcal{D}(\sigma)}(u)$ satisfy the following: $X_i(w) \leq 0$ for every $i \geq 0$, and $X_j(w) < 0$ for infinitely many $j \geq 0$, as follows from the strong law of large numbers, see e.g. [2, Theorem 22.1], and the fact that $EX_j < 0$. However, then $\mathcal{P}(CN^\sigma) = 1$. Now assume that $\mathcal{P}(R_u^\sigma > 0) > 0$. We may apply, e.g., [6, Theorem 8.3.4] and conclude that almost all $w \in \text{Run}_{\mathcal{D}(\sigma)}(u)$ satisfy $\liminf_{n \rightarrow \infty} S_n(w) = -\infty$, which implies that $\mathcal{P}(CN^\sigma) = 1$.

Now suppose that either $\mathcal{P}(R_u^\sigma < 0) > 0$, or $ER_u^\sigma \leq 0$ is not satisfied. If $\mathcal{P}(R_u^\sigma < 0) = 0$, then clearly for all $w \in \text{Run}_{\mathcal{D}(\sigma)}(u)$ and for every $n \geq 0$ we have $r(w \downarrow n) \geq -|V|$, which implies that $\mathcal{P}(CN^\sigma) = 0$. If $\mathcal{P}(R_u^\sigma < 0) > 0$ but $ER_u^\sigma > 0$, then using, e.g., [6, Theorem 8.3.4], almost all $w \in \text{Run}_{\mathcal{D}(\sigma)}(u)$ satisfy $\lim_{n \rightarrow \infty} S_n(w) = \infty$, which implies that $\mathcal{P}(CN^\sigma) = 0$.

A.4 Proof of Proposition 8

Proposition 8. *Let $A := \{v \in V \mid \text{Val}^{CN}(v) = 1\}$. Then for all $u \in V$ we have:*

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

Proof. The fact that $\max_{\tau \in MD} \mathcal{P}(\text{Reach}_A^\tau(u)) = \sup_{\tau \in HR} \mathcal{P}(\text{Reach}_A^\tau(u))$ follows from [23, Section 7.2.7], see also [7]. Clearly $\max_{\tau \in MD} \mathcal{P}(\text{Reach}_A^\tau(u)) \leq \text{Val}^{CN}(u)$. For the opposite direction, let us pick a CN-optimal MD strategy σ . Consider the Markov chain $\mathcal{D}(\sigma)$ with states V . By Lemma 19 (see Section A.3), for every BSCC C of $\mathcal{D}(\sigma)$ there is a number $x_C \in \{0, 1\}$ such that $x_C = \mathcal{P}(CN^\sigma(v)) = \text{Val}^{CN}(v)$ for all $v \in C$. Let us denote by C the union of all BSCCs C such that $x_C = 1$. Let π be a MD strategy such that $\mathcal{P}(\text{Reach}_C^\pi(u)) = \max_{\tau \in MD} \mathcal{P}(\text{Reach}_C^\tau(u))$. Then $\mathcal{P}(CN^\sigma(u)) \leq \mathcal{P}(\text{Reach}_C^\pi(u))$ because almost all runs of $\mathcal{D}(\sigma)$ eventually reach a BSCC. However, $C \subseteq A$, and thus

$$\text{Val}^{CN}(u) = \mathcal{P}(CN^\sigma(u)) \leq \mathcal{P}(\text{Reach}_C^\pi(u)) \leq \mathcal{P}(\text{Reach}_A^\pi(u)) \leq \max_{\tau \in MD} \mathcal{P}(\text{Reach}_A^\tau(u))$$

A.5 Proof of Lemma 10

We fix an arbitrary initial state s and consider the sets of strategies Σ^{MP} and Σ^{CN} defined with respect to s , see Section 3.2. Recall that a MD strategy σ in \mathcal{D} is *decreasing* if for every state u of $\mathcal{D}(\sigma)$ reachable from s there is a finite path w initiated in u such that $r(w) = -1$. We restate and prove Lemma 10 here.

Lemma 10. Σ^{CN} is the set of all decreasing strategies from Σ^{MP} .

Proof. Let σ be a MD strategy. Denote C the union of all BSCCs of $\mathcal{D}(\sigma)$ reachable from s . From [22, Theorem 1.10.2] we have that $\mathcal{P}(\text{Reach}_C^\sigma(s)) = 1$. Let $u \in C$. Similarly as in the proof of Lemma 19 (see Section A.3), we define sequences of random variables $I_1, I_2, I_3 \dots$ and X_1, X_2, \dots as follows: given a run $w \in \text{Run}_{\mathcal{D}(\sigma)}(u)$, we define $I_1(w) = 0$, and for all $n \geq 2$ we define $I_n(w)$ to be the least $m > I_{n-1}(w)$ such that $w(m) = u$. We define $X_n(w) = r(w \downarrow I_{n+1}(w)) - r(w \downarrow I_n(w))$ the reward accumulated between the n -th visit to u (inclusive) and $n + 1$ -th visit to u (non-inclusive). Observe that $X_1 = R_u^\sigma$. We define $D_n = I_{n+1}(w) - I_n(w)$. Observe that both X_1, X_2, \dots and D_1, D_2, \dots are sequences of identically distributed and independent random variables. Also EX_1 is finite, $0 < ED_1 < \infty$, and $X_1 = R_u^\sigma$ where R_u^σ is the variable defined in Lemma 19. By the strong law of large numbers, for almost all $w \in \text{Run}_{\mathcal{D}(\sigma)}(u)$

$$ER_u^\sigma = EX_1 = \lim_{n \rightarrow \infty} \frac{\sum_{i=1}^n X_i(w)}{n} = \lim_{n \rightarrow \infty} \frac{\sum_{i=1}^n X_i(w)}{\sum_{i=1}^n D_i(w)} \frac{\sum_{i=1}^n D_i(w)}{n} = \lim_{n \rightarrow \infty} \frac{\sum_{i=1}^n r(w(i))}{n} ED_1$$

Assume that $\sigma \in \Sigma^{CN}$. Let $u \in C$. We have $\mathcal{P}(CN^\sigma(u)) = 1$ because CN is prefix independent and u is reachable from s . Then, by Lemma 19, $ER_u^\sigma \leq 0$, and hence $\mathcal{P}(MP^\sigma(u)) = 1$ by the above equation. It follows that $\sigma \in \Sigma^{MP}$ because u was an arbitrary state of C , almost all runs initiated in s reach C , and MP is prefix independent.

Assume that $\sigma \in \Sigma^{MP}$ and that σ is decreasing. Let $u \in C$. We have $\mathcal{P}(MP^\sigma(u)) = 1$ because MP is prefix independent and u is reachable from s . Then, by the above equation, $ER_u^\sigma \leq 0$. Also, $\mathcal{P}(R_u^\sigma < 0) > 0$ because σ is decreasing. Hence, by Lemma 19, $\mathcal{P}(CN^\sigma(u)) = 1$. It follows that $\sigma \in \Sigma^{CN}$ because u was an arbitrary state of C , almost all runs initiated in s reach C , and CN is prefix independent.

A.6 Properties of \mathcal{D}' and the correctness of Qual-CN

Recall the MDP \mathcal{D}' from Figure 1. In this section we prove some of its properties and prove that the procedure Qual-CN from Section 3.2 is correct. Also recall that whenever we use the sets Σ^{CN} and Σ^{MP} an initial vertex s has to be specified, see the definition of the sets in Section 3.2.

Lemma 20. Let an initial vertex $s \in V$ be fixed and let $\sigma \in \Sigma^{CN}$. For every state u of $\mathcal{D}(\sigma)$ there is a finite path w of length at most $|V|^2 + 1$ initiated in u such that $r(w) = -1$.

Proof. Let w be the shortest path initiated in u such that $r(w) = -1$. Observe that if there are $i < j$ such that $w(i) = w(j)$ and $r(w \downarrow i) \leq r(w \downarrow j)$, then the path is not the shortest one (consider the path $w(0) \cdots w(i)w(j+1) \cdots$). However, then every vertex can occur at most $|V|$ times in $w \downarrow (\text{len}(w) - 1)$. This gives $|V|^2 + 1$ upper bound on the length of w .

Before we proceed to formal treatment, we briefly explain the intuition behind the construction of \mathcal{D}' . We start with explaining what information is kept in the vertices of \mathcal{D}' . In what follows, vertices of the form $(u, 1, 0)$ for some $u \in V$ are called *checkpoints*.

- First coordinate: the current vertex of \mathcal{D} ;
- second coordinate: the number by which the counter has to be decreased to make the sum of rewards gained since the last checkpoint negative;
- third coordinate: the number of steps since the last checkpoint;
- fourth coordinate, if present: the next vertex of \mathcal{D} through which the “short path” from the last checkpoint, see Lemma 20, should continue.

When the run starts, the first counter in the current vertex is 1 indicating that we wait for the sum of rewards becoming -1 , and the counter of steps is set to 0. As the play proceeds, the counters are updated accordingly. Whenever the first counter reaches value zero, the play reaches a checkpoint and the counters are reset to 1 and 0, respectively. Lemma 20 allows us to bound the (nonnegative) counters in vertices of \mathcal{D}' by $|V|^2 + 1$ and use them to make the strategy choose the right successor in transitions of the type $(u, m, n) \rightsquigarrow [u, m, n, v]$ so that v is the successor of u on the “short path” from Lemma 20. If the strategy chooses a bad successor, the player gets “punished” in terms of not satisfying the *MP* objective by entering a special vertex *div* (for *diverge*). Indeed, if the counter of the steps overflows with the accumulated reward from the last checkpoint being nonnegative, the play gets stuck in *div* and the objective *MP* is not satisfied.

Lemma 21. *Let $s \in V$ be arbitrary. The following is true.*

1. *Every MD strategy σ' in \mathcal{D}' satisfying $\mathcal{P}(MP_{\mathcal{D}',r}^{\sigma'}((s, 1, 0))) = 1$ is decreasing.*
2. *For every MD strategy σ in \mathcal{D} there is a MD strategy σ' in \mathcal{D}' such that $\mathcal{P}(CN_{\mathcal{D}',r}^{\sigma'}((s, 1, 0))) = 1$ for every $s \in V$ such that $\mathcal{P}(CN_{\mathcal{D},r}^{\sigma}(s)) = 1$.*

Proof (of 1.). First, observe that *div* is not reachable from $(s, 1, 0)$. Let (u, n, m) be a state of $\mathcal{D}' \langle \sigma' \rangle$ reachable from $(s, 1, 0)$. First, assume that $n > 0$. There is a path w from (u, n, m) to a state of the form $(u', 0, m')$, otherwise *div* would have been reachable from $(s, 1, 0)$. Let $k = \frac{|w|-1}{2}$. For every $0 \leq i \leq k$ we denote $(v_i, n_i, m_i) = w(2i)$. Then $n_0 = n > 0$ and $n_i = n + \sum_{j=0}^{i-1} r(v_j)$ for $1 \leq i \leq k$. It follows that $n + r'(w) = n + \sum_{j=0}^{k-1} r(v_j) = n_k = 0$. This implies $r'(w) < 0$.

Now assume that $n = 0$. Then $(u, n, m) \rightsquigarrow [u, 1, 0, v]$ and $[u, 1, 0, v] \rightsquigarrow (v, 1 + r(u), 1)$. Denote $w' = (u, n, m)[u, 1, 0, v]$. If $r(u) = -1$, then $r'(w' \cdot (v, 1 + r(u), 1)) = r(u) = -1$ and we are done. If $r(u) \geq 0$, then $1 + r(u) > 0$ and arguing as above we obtain a path w from $(v, 1 + r(u), 1)$ to some $(u', 0, m')$ such that $1 + r(u) + r'(w) = 0$. However, then $1 + r'(w'w) = 1 + r(u) + r'(w) = 0$ and $r'(w'w) = -1$.

Let $[u, n, m, v]$ be a state reachable from $(s, 1, 0)$. Then $n > 0$ and there is a transition $[u, n, m, v] \rightsquigarrow (v, n + r(u), m + 1)$. Arguing as above, we obtain that there is a path w from $(v, n + r(u), m + 1)$ to some $(u', 0, m')$ such that $n + r(u) + r'(w) = 0$, which implies that the $r'([u, n, m, v] \cdot w) = r(u) + r'(w) \leq -1$ and thus $r'([u, n, m, v] \cdot w') = -1$ for some prefix w' of w .

Proof (of 2.). For every $v \in V$ and $0 \leq m \leq |V|^2$ we denote by $P(v, m)$ the set of all paths in $\mathcal{D} \langle \sigma \rangle$ of length at most $|V|^2 + 1 - m$ initiated in v . We denote $\text{val}(v, m) = \min\{r(w) \mid w \in P(v, m)\}$ and $\text{val}(v, |V|^2 + 1) = 0$.

For $m \leq |V|^2$ choose $\theta(v, m) \in V$ to be an arbitrary vertex u such that $v \rightarrow u$ is a transition of $\mathcal{D}\langle\sigma\rangle$ and

$$\text{val}(u, m+1) = \min\{\text{val}(u', m+1) \mid v \rightarrow u' \text{ in } \mathcal{D}\langle\sigma\rangle\}$$

Let us define a strategy σ' as follows:

- Let $(u, n, m) \in V'_N$.
 - If $m = |V|^2 + 1$ and $n > 0$, we put $\sigma'((u, n, m)) = \text{div}$
 - If $m \leq |V|^2 + 1$ and $n = 0$, we put $\sigma'((u, n, m)) = [u, 1, 0, \theta(u, 0)]$
 - If $m < |V|^2 + 1$ and $n > 0$, we put $\sigma'((u, n, m)) = [u, n, m, \theta(u, m)]$
- For every $[u, n, m, v] \in V'_N$, we put $\sigma'([u, n, m, v]) = (v, n + r(u), m + 1)$

Fix an arbitrary $s \in V$ such that $\mathcal{P}(\text{CN}_{\mathcal{D},r}^\sigma(s)) = 1$. Denote by R the set of all states of the form (u, n, m) reachable from $(s, 1, 0)$. We prove that $n + \text{val}(u, m) \leq 0$ for all $(u, n, m) \in R$ by induction on m . If $m = 0$, then $n = 1$ and Lemma 20 implies $\text{val}(u, 0) \leq -1$.

Consider $(u, n, m) \in R$ such that $m > 0$. Then $(u', n', m') \rightarrow [u', n'', m'', u] \rightarrow (u, n, m)$ in $\mathcal{D}\langle\sigma\rangle$ for some $(u', n', m') \in R$. Now either $n'' = 1$ and $m'' = 0$, or $n'' = n'$ and $m'' = m'$. First, assume that $n'' = 1$ and $m'' = 0$. Then $n = 1 + r(u')$ and $m = 1$. By Lemma 20, $\text{val}(u', 0) \leq -1$ and thus by definition of σ' , $1 + r(u') + \text{val}(u, 1) \leq 0$. Now assume that $n'' = n'$ and $m'' = m'$. Then $n = n' + r(u')$ and $m = m' + 1$. By induction hypothesis, $n' + \text{val}(u', m') \leq 0$, and thus by definition of σ' , $n' + r(u') + \text{val}(u, m) \leq n' + \text{val}(u', m') \leq 0$.

This proves that if $(u, n, |V|^2 + 1) \in R$, then $n = 0$. It follows that div is not reachable. Given $[u, n, m, v] \in V'$, we define $\Theta([u, n, m, v]) = u$. Given $w \in \text{Run}_{\mathcal{D}\langle\sigma'\rangle}((s, 1, 0))$, we define a run $\Theta(w) \in \text{Run}_{\mathcal{D}\langle\sigma\rangle}(s)$ by $\Theta(w)(k) = \Theta(w(2k+1))$. We have that Θ induces an isomorphism of the probability spaces $\text{Run}_{\mathcal{D}\langle\sigma'\rangle}((s, 1, 0))$ and $\text{Run}_{\mathcal{D}\langle\sigma\rangle}(s)$. Indeed, it follows from the following three facts: First, div is not reachable. Second, if $u \in V_N$ and $[u, n, m, v] \in R$, then $\sigma(u) = v$ and $[u, n, m, v] \xrightarrow{1} (v, n'', m'') \xrightarrow{1} [v, n', m', v']$ in $\mathcal{D}\langle\sigma'\rangle$ for some n'', m'', n', m', v' . Third, if $u \in V_P$ and $[u, n, m, v] \in R$, then $[u, n, m, v] \xrightarrow{x} (u', n'', m'') \xrightarrow{1} [u', n', m', v']$ in $\mathcal{D}\langle\sigma'\rangle$ for some n'', m'', n', m', v' iff $u \hookrightarrow u'$ is assigned x in \mathcal{D} . Also, $w \in \text{CN}_{\mathcal{D},r'}^{\sigma'}((s, 1, 0))$ iff $\Theta(w) \in \text{CN}_{\mathcal{D},r}^\sigma(s)$. Thus $\mathcal{P}(\text{CN}_{\mathcal{D},r}^\sigma(s)) = \mathcal{P}(\text{CN}_{\mathcal{D},r'}^{\sigma'}((s, 1, 0))) = 1$.

So far we have that, for a fixed initial vertex $s \in V$, if $\Sigma^{\text{CN}} \neq \emptyset$ in \mathcal{D} then $\Sigma^{\text{CN}} = \Sigma^{\text{MP}} \neq \emptyset$ in \mathcal{D}' . It remains to prove the other implication. We do this in two steps and we need the following notion:

Definition 22. A deterministic strategy σ in \mathcal{D} is said to be finite-memory (FD) if there is a deterministic finite automaton (DFA) \mathcal{A} such that for every $wu \in V^*V$ the value of $\sigma(wu)$ depends only on u and the current state k of \mathcal{A} after reading w (we write $\sigma(u, k)$ instead of $\sigma(wu)$).

Lemma 23. Given a MD strategy σ' in \mathcal{D}' there is a FD strategy σ in \mathcal{D} computable in polynomial time such that $\mathcal{P}(\text{CN}_{\mathcal{D},r}^\sigma(s)) = 1$ for every $s \in V$ with $\mathcal{P}(\text{CN}_{\mathcal{D}',r'}^{\sigma'}((s, 1, 0))) = 1$.

Proof. Let us define σ as follows: Let $\mathcal{A} = (K, V, \zeta, k_{in})$ where

- K consists of all vertices of V' of the form $[u, n, m, v]$.
- ζ is defined as follows: Let $[u, n, m, v] \in K$ and $u' \in V$. If $u \hookrightarrow u'$, then we define $\zeta([u, n, m, v], u')$ to be the unique vertex of the form $[u', n', m', v']$ satisfying $[u, n, m, v] \rightarrow (u', n'', m'') \rightarrow [u', n', m', v']$ in $\mathcal{D}\langle\sigma'\rangle$ for some n'' and m'' . Otherwise, we define $\zeta([u, n, m, v], u')$ to be an arbitrary state of \mathcal{A} .
- Define $k_{in} = \sigma'((s, 1, 0))$.

For $u \in V_N$, we define $\sigma(u, [u, n, m, v]) = v$. For $u' \neq u$ we define $\sigma(u', [u, n, m, v])$ to be an arbitrary vertex u'' such that $u' \hookrightarrow u''$.

The rest is similar to the end of the proof of Lemma 21. Given $[u, n, m, v] \in V'$, we define $\Theta([u, n, m, v]) = u$. Given $w \in \text{Run}_{\mathcal{D}'(\sigma')}((s, 1, 0))$, we define a run $\Theta(w) \in \text{Run}_{\mathcal{D}(\sigma)}(s)$ by $\Theta(w)(k) = \Theta(w(2k + 1))$. Then Θ induces an isomorphism of the probability spaces $\text{Run}_{\mathcal{D}'(\sigma')}((s, 1, 0))$ and $\text{Run}_{\mathcal{D}(\sigma)}(s)$. Indeed, it follows from the following facts: First, div is not reachable from $(s, 1, 0)$ in $\mathcal{D}'(\sigma')$. Second, if $[u, n, m, v] \in V'_N$, then $[u, n, m, v] \xrightarrow{1} (v, n'', m'') \xrightarrow{1} [v, n', m', v']$ in $\mathcal{D}'(\sigma')$ for some n'', m'', n', m', v' . Third, for $[u, n, m, v] \in V'_P$, $[u, n, m, v] \xrightarrow{x} (u', n'', m'') \xrightarrow{1} [u', n', m', v']$ for some n'', m'', n', m', v' iff the transition $u \hookrightarrow v$ is assigned x in \mathcal{D} . Also, $w \in \text{CN}_{\mathcal{D}', r}^{\sigma'}((s, 1, 0))$ iff $\Theta(w) \in \text{CN}_{\mathcal{D}, r}^{\sigma}(s)$. Thus $\mathcal{P}(\text{CN}_{\mathcal{D}, r}^{\sigma}(s)) = \mathcal{P}(\text{CN}_{\mathcal{D}', r}^{\sigma'}((s, 1, 0))) = 1$.

Remark 24. Since the DFA \mathcal{A} in the proof of Lemma 23 effectively simulates the Markov Chain \mathcal{M} , we will simplify the notation used in the procedure CN-FD-to-MD by identifying the MD strategy for \mathcal{D}' with its associated FD strategy for \mathcal{D} .

Lemma 25. *Let σ' be a FD strategy in \mathcal{D} . Then the procedure CN-FD-to-MD computes in time polynomial in the size of the DFA associated with σ' a MD strategy σ such that $\mathcal{P}(\text{CN}^{\sigma}(s)) = 1$ for every $s \in V$ with $\mathcal{P}(\text{CN}_{\mathcal{D}, r}^{\sigma'}(s)) = 1$.*

Proof. Denote \mathcal{A} the DFA associated with σ' . Further denote K the set of its states, \hat{k} its initial state, ζ its transition function. Recall that the input alphabet of such an automaton is V , the set of vertices of the MDP \mathcal{D} . We combine \mathcal{A} with \mathcal{D} and σ' by means of parallel sequential composition into a finite Markov chain \mathcal{M} . More precisely, the set of vertices of \mathcal{M} is the set $V \times K$ and the transitions and probabilities are defined as follows: For $u \in V_N$ and $k \in K$ we put $(u, k) \xrightarrow{1} (u', k')$ if and only if $\sigma'(u, k) = u'$ and $k' = \zeta(k, u)$. For $u \in V_P$ and $k \in K$ we put $(u, k) \xrightarrow{x} (u', k')$ if and only if $u \hookrightarrow u'$ is assigned the probability x and $k' = \zeta(k, u)$. Given $(u, k) \in V \times K$, we denote the projection $\pi_1((u, k)) = u$ and define $r'((u, k)) = r(u)$.

The following procedure CN-FD-to-MD computes a sequence of Markov chains M_n , $0 \leq n \leq |V|$ with state spaces $V \times K$, transitions \rightarrow_n and probabilities Prob_n . Then it extracts the strategy σ from the last M_n , $n = |V|$. For every $0 \leq n \leq |V|$, let C_n be a union of all BSCCs of M_n reachable from (s, \hat{k}) . We say that $u \in V$ is *ambiguous in C_n* if for at least two $k_1, k_2 \in K$, $k_1 \neq k_2$, both $(u, k_1), (u, k_2) \in C_n$. For every M_n an an initial vertex (u, k) we define a random variable $R_{(u, k)}$ as follows: given a run w we set $S = \{m > 0 \mid \pi_1(w(m)) = u\}$ and put

$$R_{(u, k)}(w) = \begin{cases} r'(w \downarrow m) & S \neq \emptyset, m = \min S \\ \perp & S = \emptyset \end{cases}$$

Since every M_n is finite, $R_{(u, k)}$ is almost surely defined whenever (u, k) lies $s \in V$ with $\mathcal{P}(\text{CN}_{\mathcal{D}, r}^{\sigma'}(s)) = 1$ in a BSCC, and the expectation $ER_{(u, k)}$ is finite, see [22, Theorem 1.10.2].

Procedure CN-FD-to-MD computes σ , we first estimate its running time. The while-loop on line 2 is executed at most $|V|$ -times because every state (u_a, k) picked in step 3 is no longer ambiguous in later iterations. We show that the picking in step 3 takes polynomial time. First, due to, e.g. [12, XV.7], $ER_{(u_a, k)}$ can be expressed as a unique solution of a linear system of equations, computable in polynomial time. So we can compute $ER_{(u_a, k)}$ in polynomial time and check whether $ER_{(u_a, k)} \leq 0$. Second, the problem whether $\mathcal{P}(R_{(u_a, k)} < 0) > 0$ is equivalent to the existence of a negative weighted cycle in the BSCC containing (u_a, k) , which can be decided in polynomial time using, e.g., the Bellman-Ford algorithm. Time complexity of the procedure Max-Reach on line 9 has already been analyzed in Section 3.1.

Let us prove correctness. Fix some $s \in V$ with $\mathcal{P}(\text{CN}_{\mathcal{D}, r}^{\sigma'}(s)) = 1$. We prove by induction that for all h , $0 \leq h \leq |V|$: $\mathcal{P}(\text{CN}((s, \hat{k}))) = 1$ in M_h . For $h = 0$ this is true by the choice of s . Assume that the statement is

Procedure CN-FD-to-MD(σ') – computing a MD CN-optimal strategy from a FD one.

Data: The product Markov Chain \mathcal{M} determined by the strategy σ' .

Result: Produce a CN-optimal MD strategy σ .

```

1  $n \leftarrow 0, M_0 \leftarrow \mathcal{M}$ 
2 while there are states ambiguous in  $C_n$  do
3   Pick  $(u_a, k) \in C_n$ , such that  $u_a$  is ambiguous in  $C_n$ ,  $ER_{(u_a, k)} \leq 0$ , and  $\mathcal{P}_{(u_a, k)}(R < 0) > 0$ .
4   Compute  $M_{n+1}$  from  $M_n$  as follows:
5   Set  $(v, k') \xrightarrow{x}_{n+1}(u_a, k)$  iff  $(v, k') \xrightarrow{x}_n(u_a, k')$  for some  $k'$ .
6   Set  $(v, k) \xrightarrow{x}_{n+1}(u, k')$  iff  $(v, k) \xrightarrow{x}_n(u, k')$  for  $u \neq u_a$ .
7    $n \leftarrow n + 1$ 
8  $C \leftarrow \{u \in V \mid (u, k) \in C_n\}$ .
9  $(\varrho, -) \leftarrow \text{Max-Reach}(\mathcal{D}, C)$ 
10 for  $u \in V$  do if  $u \notin C$  then  $\sigma(u) = \varrho(u)$  else  $\sigma(u) = v$  where  $\exists k, k' \in K : (u, k) \rightarrow_n(v, k')$ 
11 return  $\sigma$ 

```

true for some $h = n \in \mathbb{N}_0$, we prove it for $h = n + 1$. First, we prove that if there is some v ambiguous in C_n , then there is $(v, k) \in C_n$ such that $ER_{(v, k)} \leq 0$ and $\mathcal{P}(R_{(v, k)} < 0) > 0$. Let C be a BSCC of M_n reachable from (s, \hat{k}) and containing at least two states from $\{v\} \times K$. Let us denote $C^v := (\{v\} \times K) \cap C = \{(v, k_1), \dots, (v, k_\ell)\}$.

We define sequences of random variables $I_0, I_1, I_2 \dots$ and X_1^i, X_2^i, \dots where $i \in \{1, \dots, \ell\}$ as follows: Let w be a run in M_n initiated in some $(v, k_j) \in C^v$. We define $I_0(w) = 0$, and for all $j \geq 1$ we define $I_j(w)$ to be the least $m > I_{j-1}(w)$ such that $w(m) \in C^v$. Let $i \in \{1, \dots, \ell\}$ and let m_1, m_2, m_3, \dots be all indexes such that $I_{m_j}(w) = (v, k_i)$. We define $X_j^i(w) = r'(w \downarrow I_{m_{j+1}}(w)) - r'(w \downarrow I_{m_j}(w))$ the reward accumulated between the j -th visit to (v, k_i) and next visit to C^v .

Consider the Markov chain M_n . Observe that EX_1^i is independent of the actual initial vertex $(v, k_j) \in C^v$ and that $EX_1^i = ER_{(v, k_j)}$. Also, for a fixed $i, 1 \leq i \leq \ell$, the variables $X_j^i, j \geq 1$ are independent and identically distributed. We claim that $EX_1^i \leq 0$ for some i and $\mathcal{P}(X_1^i < 0) > 0$. Assume, to the contrary, that there is no such i and let us denote $B = \{i \mid EX_1^i > 0\}$. The variables X_j^i generate ℓ random walks of the form S_1^i, S_2^i, \dots by $S_n^i = \sum_{j=1}^n X_j^i$. For every $i \in B$ the walk S_j^i drifts almost surely to ∞ , by, e.g., [6, Theorem 8.3.4]. On the other hand, for every $i \in \{1, \dots, \ell\} \setminus B$ the walk never reaches values smaller than a fixed number. Since for almost all runs starting in some $(v, k_j) \in C^v$ we have $\liminf_{n \rightarrow \infty} r(w \downarrow n) = \liminf_{n \rightarrow \infty} \sum_{i=1}^{\ell} S_n^i$, it follows that in M_n : $\mathcal{P}(CN((v, k_j))) = 0$, and hence $\mathcal{P}(CN((s, \hat{k}))) < 1$, a contradiction.

Now we prove that in M_{n+1} : $\mathcal{P}_{(s, \hat{k})}(CN) = 1$. Assume that (u_a, k) is the state selected in step 3. Then the expected value $ER_{(u_a, k)}$ is the same in both M_{n+1} and M_n and thus not positive. Further $\mathcal{P}(R_{(u_a, k)} < 0) > 0$ in M_{n+1} and no states of the form (u_a, k') where $k' \neq k$ are reachable from (u_a, k) in M_{n+1} . By Lemma 19, $\mathcal{P}(CN((u_a, k))) = 1$ in M_{n+1} . Let A be the set of all runs of $\text{Run}_{M_n}((s, \hat{k}))$ not reaching (u_a, k) . Clearly, the probability of A is the same in M_n as in M_{n+1} . Hence, $\mathcal{P}(CN((s, \hat{k}))) = 1$ in M_{n+1} .

Finally, note that for every n the Markov chain M_n has the following properties:

- For each $(u, k) \in V_P \times K$, if $(u, k) \xrightarrow{x}(v, k')$, then $u \xrightarrow{c^x} v$.
- For each $(u, k) \in V_N \times K$, if $(u, k) \xrightarrow{x}(v, k')$, then $x = 1$ and $u \xrightarrow{c} v$.

Hence, in step 8 the set C is reachable from s with probability 1 using a suitable MD strategy ϱ , line 9. Consequently the strategy σ for \mathcal{D} is well-defined (line 10) and satisfies $\mathcal{P}(CN^{\mathcal{D}}(s)) = 1$.

The correctness of the reduction represented by the procedure Qual-CN follows from Lemma 21, Lemma 23, Remark 24, and Lemma 25.

A.7 Correctness of Qual-MP

Denote $W = \{s \in V \mid \exists \text{ MD strategy } \sigma : \mathcal{P}(MP^\sigma(s)) = 1\}$. In this section we prove that the set A and MD strategy σ computed by the procedure Qual-MP satisfy: $\forall s \in W : \mathcal{P}(MP^\sigma) = 1$ and $W = A$.

Choose an arbitrary $s \in V$. Let σ be a MD strategy. Let us denote $BSCC[\mathcal{D}(\sigma)]$ the set of all BSCCs of $\mathcal{D}(\sigma)$ reachable from s . By standard arguments from the theory of Markov chains (see e.g. [22, Section 1.5]), $\sum_{C \in BSCC[\mathcal{D}(\sigma)]} \mathcal{P}(Reach_C^\sigma(s)) = 1$. Recall also the random variable $V[\sigma, r]$ defined in Section 3.3. In particular recall that [22, Theorem 1.10.2] implies that for almost all runs w $V[\sigma, r](w) = \lim_{n \rightarrow \infty} \frac{r(w \downarrow n)}{n}$. Moreover, using [22, Theorem 1.10.2] again, for every $C \in BSCC[\mathcal{D}(\sigma)]$ there is a constant $a_C \in \mathbb{R}$ such that $V[\sigma, r] = a_C$ almost surely on the condition of hitting C . Thus for the expected value we have

$$EV[\sigma, r] = \sum_{C \in BSCC[\mathcal{D}(\sigma)]} a_C \cdot \mathcal{P}(Reach_C^\sigma(s))$$

We prove that there is a MD strategy ϱ computable in polynomial time such that $EV[\varrho, r] = \min_\sigma EV[\sigma, r]$ where the minimum is taken over MD strategies.

Let σ be a MD strategy. We define a sequence of random variables $V_1[\sigma, r], V_2[\sigma, r], \dots$ such that $V_n[\sigma, r] = r(w \downarrow n)$ for every run $w \in Run_{\mathcal{D}(\sigma)}(u_0)$ and every $n \geq 1$. Let us denote $EV_n[\sigma, r]$ the expected value of $V_n[\sigma, r]$ (i.e. $EV_n[\sigma, r] = \sum_{i=-n}^n i \cdot \mathcal{P}(V_n[\sigma, r] = i)$).

Note that

$$\frac{EV_n[\sigma, r]}{n} = \frac{\sum_{i=-n}^n i \cdot \mathcal{P}(V_n[\sigma, r] = i)}{n} = \sum_{i=-n}^n \frac{i}{n} \cdot \mathcal{P}\left(\frac{V_n[\sigma, r]}{n} = \frac{i}{n}\right) = E \frac{V_n[\sigma, r]}{n}$$

and that $|\frac{V_n[\sigma, r]}{n}| \leq 1$. Hence by the dominated convergence theorem (see e.g. [2, Theorem 16.4])

$$\lim_{n \rightarrow \infty} \frac{EV_n[\sigma, r]}{n} = \lim_{n \rightarrow \infty} E \frac{V_n[\sigma, r]}{n} = EV[\sigma, r]$$

Using either [13, Theorem 2.9.4], or [23, Theorem 9.3.8], and a P-time algorithm for linear programming, one can construct a polynomial time algorithm which computes a MD strategy ϱ such that (taking the minima over MD strategies)

$$EV[\varrho, r] = \lim_{n \rightarrow \infty} \frac{EV_n[\varrho, r]}{n} = \min_\sigma \lim_{n \rightarrow \infty} \frac{EV_n[\sigma, r]}{n} = \min_\sigma EV[\sigma, r]$$

and also computes the value $EV[\varrho, r]$.

In the proof of correctness and the complexity estimates of Qual-MP we will denote r_i , T_i , and A_i the reward represented by \hat{r} , the content of the set T , and the set A , respectively, before the i -th iteration of the while-loop, in particular $r_0 = r$, $T_0 = \emptyset$, and $A_0 = \emptyset$. We also denote ϱ_i the strategy ϱ from line 5 computed in the i -th iteration of the while-loop.

Choose some $s \in W$ so that there is a strategy $\bar{\sigma}$ such that $\mathcal{P}(MP^{\bar{\sigma}}(s)) = 1$, i.e., $V[\bar{\sigma}, r](w) \leq 0$ almost surely. Given i , we define a MD strategy σ^i such that for every $u \in V$

$$\sigma^i(u) = \begin{cases} v & (u, v) \in T_i \\ \bar{\sigma}(u) & \text{otherwise.} \end{cases}$$

The algorithm keeps the following invariants:

- (a) $V[\sigma^i, r_i](w) \leq 0$ and $V[\sigma^i, r](w) \leq 0$ almost surely.

- (b) For every $u \in V \setminus A_i$ and every strategy σ in \mathcal{D} , the probability of reaching A_i from u is strictly less than 1. There is no path from any state of A_i to $V \setminus A_i$ in $\mathcal{D}(\sigma^i)$.
- (c) A and $V_?$ are disjoint.

The invariant (c) follows by an easy induction from lines 1 and 12.

Clearly, the invariant (a) implies that on line 5 the strategy ϱ always exists. We prove that a BSCC C from line 6 exists. Note that by the invariant (b), for all $C \in \text{BSCC}[\mathcal{D}(\sigma)]$ either $C \cap A_i = \emptyset$, or $C \subseteq A_i$, and there must be at least one C such that $C \cap A_i = \emptyset$, otherwise s could not have been in A_i , contradicting line 3 and the invariant (c). Also there are numbers $a_{C,i}$ for every $C \in \text{BSCC}[\mathcal{D}(\varrho_i)]$ such that $V[\varrho_i, r_i] = a_{C,i}$ almost surely on the condition of hitting C , and

$$EV[\varrho_i, r_i] = \sum_{C \in \text{BSCC}[\mathcal{D}(\varrho_i)]} a_{C,i} \cdot \mathcal{P}(\text{Reach}_C^{\varrho_i}(s))$$

However, all $C \in \text{BSCC}[\mathcal{D}(\varrho_i)]$ such that $C \subseteq A_i$ satisfy $a_{C,i} = 0$. Hence, there must be at least one $C_{wit} \in \text{BSCC}[\mathcal{D}(\varrho_i)]$ such that $C_{wit} \cap A_i = \emptyset$ and $a_{C_{wit},i} > 0$.

Now every $D \in \text{BSCC}[\mathcal{D}(\sigma^{i+1})]$ satisfies either $D = C_{wit} \subseteq A_{i+1} \setminus A_i$, or $D \subseteq (V \setminus A_{i+1}) \cup A_i$ and $D \in \text{BSCC}[\mathcal{D}(\sigma^i)]$. Moreover, transitions between states of C_{wit} in $\mathcal{D}(\sigma^{i+1})$ coincide with transitions between states of C_{wit} in $\mathcal{D}(\sigma^i)$. Also, transitions between states of every $D \neq C_{wit}$ in $\mathcal{D}(\sigma^{i+1})$ coincide with transitions between states of D in $\mathcal{D}(\sigma^i)$.

Then almost all $w \in \text{Reach}_{C_{wit}}^{\sigma^{i+1}}(s)$ satisfy $V[\sigma^{i+1}, r_{i+1}](w) \leq 0$ because r_{i+1} assigns 0 to all states of C_{wit} . Also, almost all $w \in \text{Reach}_D^{\sigma^{i+1}}(s)$ where $D \neq C_{wit}$ satisfy $V[\sigma^{i+1}, r_{i+1}](w) = V[\sigma^i, r_i](w) \leq 0$ due to the invariant (a) for i . It follows that $V[\sigma^{i+1}, r_{i+1}](w) \leq 0$ for almost all runs $w \in \text{Run}_{\mathcal{D}(\sigma^{i+1})}(s)$.

Moreover, almost all $w \in \text{Reach}_{C_{wit}}^{\sigma^{i+1}}(s)$ satisfy $V[\sigma^{i+1}, r](w) \leq 0$ because r_i coincides with r on C_{wit} and almost all runs $w \in \text{Reach}_{C_{wit}}^{\sigma^i}(s)$ satisfy $V[\sigma^i, r_i](w) \leq 0$. Also, almost all $w \in \text{Reach}_D^{\sigma^{i+1}}(s)$ where $D \neq C_{wit}$ satisfy $V[\sigma^{i+1}, r](w) = V[\sigma^i, r](w) \leq 0$ due to the invariant (a) for i . Hence, for almost all runs $w \in \text{Run}_{\mathcal{D}(\sigma^{i+1})}(s)$ we have $V[\sigma^{i+1}, r](w) \leq 0$.

It follows that the invariant (a) is preserved. The invariant (b) is preserved due to computation of τ on line 7, A' on line 8 and update of A in line 10. Finally, the strategy σ defined on line 13 has the desired properties because it coincides with σ^{i+1} on all reachable states, and σ^{i+1} satisfies the invariant (a). This also implies that the vertex s was put into A' on line 8 and consequently to A on line 10 in some iteration of the while-loop. Thus $W \subseteq A$. Since by arguments similar as above we can show that for every $s \in A$ we have $\mathcal{P}(\text{MP}^\sigma(s)) = 1$ the correctness is proved.

Let us now consider the complexity. By [22, Theorem 1.10.2], for every $C \in \text{BSCC}[\mathcal{D}(\varrho_i)]$ the constant $a_{C,i} \in \mathbb{R}$ defined above is equal to $\sum_{u \in C} \mu(u) \cdot r_i(u)$, where μ is the invariant distribution for C (note that C can be considered as a standalone irreducible Markov chain within $\mathcal{D}(\varrho_i)$), which is a unique solution of a system of linear equations, and thus computable in polynomial time. Hence, a suitable BSCC satisfying the conditions from line 6 can be computed in polynomial time. In Section 3.3 we already showed that the strategy ϱ from line 5 can be found in polynomial time. In Section 3.1 we showed that also finding the strategy τ on line 7 can be done in polynomial time. Other steps can be clearly taken in polynomial time. Since the set A grows with every iteration of the while-loop by at least one vertex, the loop itself is executed at most $|V|$ -times. Thus the procedure `Qual-MP` runs in polynomial time.

B Proofs of Section 4

For the rest of this section, we fix an OC-MDP $\mathcal{A} = (Q, \delta^{=0}, \delta^{>0}, (Q_N, Q_P), P^{=0}, P^{>0})$ and a non-empty set $F \subseteq Q$ of final states. We assume (without restrictions) that for each $q \in F$, the configuration $q(0)$ has only one outgoing transition $q(0) \mapsto q(0)$. We also use N to denote $2^{|Q|}$.

Obviously, $OptValOne^{NT} \subseteq ValOne^{NT}$ and $OptValOne^{ST} \subseteq ValOne^{ST}$, but it is not immediately clear whether the inclusions are proper. As we shall see, the sets $OptValOne^{NT}$, $ValOne^{NT}$, and $OptValOne^{ST}$ have a regular structure which can be captured by finite state automata, and optimal strategies are either counter-oblivious or counter-regular.

Definition 26 (regular sets of configurations, counter-regular strategies). An \mathcal{A} -automaton is a pair (M, ϱ) where $M = (C, \{a\}, \gamma, F)$ is a deterministic finite-state automaton and $\varrho : Q \rightarrow C$ a mapping. A set of configurations of \mathcal{A} recognized by (M, f) consists of all $p(i) \in Q \times \mathbb{N}_0$ such that M accepts the word a^i from the initial state $\varrho(p)$. A set of configurations is regular if it is recognized by some \mathcal{A} -automaton.

A MD strategy σ is counter-regular if there is an \mathcal{A} -automaton (M, ϱ) and a function $f : Q \times C \rightarrow \delta^{>0}$, where C is the set of states of M , such that for all $p(i) \in Q \times \mathbb{N}$ we have that $\sigma(p(i)) = f(p, q)$, where $q \in C$ is the state entered from $\varrho(p)$ after reading the word a^i .

We start by proving the results about NT objectives.

Theorem 12. The sets $ValOne^{NT}$ and $OptValOne^{NT}$ are equal. 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 in every configuration of $ValOne^{NT} = OptValOne^{NT}$.

Proof. We start by showing that for all $i \geq |Q|$ and all $p \in Q$ such that $p(i) \in ValOne^{NT}$ we have that

$$1 = \sup_{\tau \in HR} \mathcal{P}(NT^\tau(p(i))) = \sup_{\tau \in HR} \mathcal{P}(CN_{\mathcal{A}}^\tau(p(i))) \quad (1)$$

Let us fix some $p(i) \in Q \times \mathbb{N}_0$ where $i \geq |Q|$. Consider an arbitrary HR strategy τ for $\mathcal{D}_{\mathcal{A}}^\rightarrow$. For every $0 \leq j \leq i$, we define the set $U_j^\tau \subseteq Q$ which consists of all $q \in Q$ such that with probability > 0 a run from $p(i)$ under τ visits $q(j)$ before visiting any other configuration $s(k)$ with $k \leq j$. Consider further an arbitrary infinite sequence $\varepsilon_1, \varepsilon_2, \dots$ of positive reals where $\lim_{n \rightarrow \infty} \varepsilon_n = 0$, and an infinite sequence of strategies $\sigma_1, \sigma_2, \dots$ such that $\mathcal{P}(NT^{\sigma_j}(p(i))) \geq 1 - \varepsilon_j$ for all j . Since there are only finitely many collections of $i + 1$ subsets of Q , there are subsequences $\varepsilon_{d_1}, \varepsilon_{d_2}, \dots$ and $\sigma_{d_1}, \sigma_{d_2}, \dots$, and a collection $U_0, \dots, U_i \subseteq Q$ such that $\lim_{n \rightarrow \infty} \varepsilon_{d_n} = 0$, $\mathcal{P}(NT^{\sigma_{d_j}}(p(i))) \geq 1 - \varepsilon_{d_j}$ for all j , and moreover $U_k = U_k^{\sigma_{d_j}}$ for all j and all $0 \leq k \leq i$.

Since $i + 1 > |Q|$, there must be some k , where $0 \leq k \leq i$, such that $U_k \subseteq \bigcup_{i \geq j > k} U_j$. Thus, for every $q \in U_k$ and $l \in \mathbb{Z}$ the strategies σ_{d_j} , $j \geq 1$ induce strategies in $\mathcal{D}_{\mathcal{A}}^{\leftrightarrow}$ for reaching $U_k \times \{l - 1, l - 2, \dots\}$ from $q(l)$ with a probability arbitrarily close to 1. This allows us to construct strategies for satisfying $CN_{\mathcal{A}}$ with probability arbitrarily close to 1 from every $q(l)$, $q \in U_k$, $l \in \mathbb{Z}$. Indeed, for an arbitrary $\delta > 0$ consider the sequence $\{\delta_j\}_{j=1}^\infty$, where $\delta_j = \delta \cdot 2^{-j}$. For every $w \in (Q \times \mathbb{Z})^+$ which starts with some $q(l) \in U_k \times \mathbb{Z}$ we denote *min-step* every index j such that

- $w(j) = q(m)$ for some $q \in U_k$, $m \in \mathbb{Z}$,
- for all h such that $0 \leq h < j$ we have that if $w(h) = q(m')$, then $q \notin U_k$ or $m' > m$.

We define a strategy τ by setting $\tau(w) = \tau_j(w')$ where j is the number of min-steps in w , $w' = w(m) \cdots w(|w|-1)$ with m being the last min-step, and τ_j is a δ_j -optimal strategy for satisfying $CN_{\mathcal{A}}$ from $w(m)$. It follows that $\mathcal{P}(CN_{\mathcal{A}}^{\tau}(q(l))) \geq \prod_{j=1}^{\infty} 1 - \delta_j \geq 1 - \delta$. Since the strategies σ_{d_j} also induce strategies for reaching $U_k \times \mathbb{Z}$ from $p(i)$ with probability arbitrarily close to 1, we proved (1).

By applying Theorem 4, we can conclude that our theorem is true for all configurations of the form $p(i)$ with $p \in Q$, $i \geq |Q|$, since an optimal CMD strategy for $CN_{\mathcal{A}}$ induces directly an optimal CMD strategy in $\mathcal{D}_{\mathcal{A}}^{\rightarrow}$ for NT . Let us denote this strategy by σ .

Consider now the case $p(i)$ when $i < |Q|$. Let

$$A = \text{ValOne}^{NT} \cap \{q(j) \mid q \in Q, j \geq |Q|\}$$

Consider a finite MDP \mathcal{D} with vertices $Q \times \{0, 1, \dots, |Q|\}$ such that for all $q \in Q$ the vertices $q(|Q|)$ are stochastic with only one transition $q(|Q|) \xrightarrow{1} q(|Q|)$ and the rest is just restriction of transitions and probabilities from $\mathcal{D}_{\mathcal{A}}^{\rightarrow}$. Then the following is equivalent due to standard results for finite MDP (see e.g. [7]):

- $p(i) \in \text{ValOne}^{NT}$
- There are strategies in $\mathcal{D}_{\mathcal{A}}^{\rightarrow}$ for reaching $A \cup (Q \times \{0\})$ from $p(i)$ with probability arbitrarily close to 1.
- There are strategies in \mathcal{D} for reaching $(A \cap (Q \times \{|Q|\})) \cup (Q \times \{0\})$ from $p(i)$ with probability arbitrarily close to 1.
- There is a MD strategy τ in \mathcal{D} computable in polynomial time for reaching $(A \cap (Q \times \{|Q|\})) \cup (Q \times \{0\})$ from $p(i)$ with probability 1.
- $p(i) \in \text{OptValOne}^{NT}$ with the witnessing strategy being τ extended with σ for configurations $q(m)$, $m \geq |Q|$.

We have already defined a CMD strategy σ such that $\mathcal{P}(NT^{\sigma}(p(i))) = 1$ for all $i \in \mathbb{N}$ and p such that $\{p\} \times \mathbb{N} \subseteq \text{ValOne}^{NT}$ (call these p safe). To finish the proof of our theorem, it remains to redefine σ for configurations $p(i)$, $p \in Q$, $i < |Q|$ such that $p(i) \in \text{ValOne}^{NT}$ but $p(|Q|) \notin \text{ValOne}^{NT}$ (call these p unsafe). Note that due to (1) every $p \in Q$ is either safe or unsafe. For every unsafe p there is some $i_p < |Q|$ such that $p(i) \in \text{ValOne}^{NT}$ iff $i \leq i_p$. Take the MD strategy τ such that $\mathcal{P}(NT^{\tau}(p(i_p))) = 1$. Note that this strategy can be chosen one for all such $p(i_p)$. We now redefine the CMD strategy σ by redefining its selector f : $f(p)$ is the rule generating the transition chosen by τ in $p(i_p)$. Since no configuration with an unsafe state is reached from a configuration with a safe state under σ this does not influence the property that $\mathcal{P}(NT^{\sigma}(p(i))) = 1$ for all safe p . Moreover from the definition of f and the choice of i_p , almost all runs from $p(i)$, $i \leq i_p$ under σ either visit a configuration with a safe state or a configuration from $Q \times \{0\}$ or $q(i_q + i - i_p)$ with q unsafe. Thus by double induction, first on $|Q| - i_p$ then on i , for all unsafe p and $i \leq i_p$ we have $\mathcal{P}(NT^{\sigma}(p(i))) = 1$.

Since $\text{ValOne}^{NT} = \{(q, i) \mid q \text{ is safe, } i \in \mathbb{N}\} \cup \{(q, i) \mid q \text{ is unsafe, } i \leq i_q\}$, we have proved the theorem. \square

Remark 27. Let $I = \{p \in Q \mid p(i) \in \text{ValOne}^{NT} \text{ for all } i \in \mathbb{N}\}$. Then

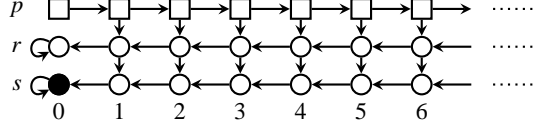
- for every $q \in I \cap Q_P$ we have that if $(q, c, q') \in \delta^{>0}$, then $q' \in I$;
- if $q \in I \cap Q_N$, then there is $(q, c, q') \in \delta^{>0}$ such that $q' \in I$.

This means that we can define a OC-MDP \mathcal{A}_I obtained from \mathcal{A} by

- restricting the set of control states to I ;
- restricting the set of positive rules to the rules of the form (q, c, q') where $q, q' \in I$ and the probability assignment is preserved;
- redefining the set of zero rules to $\{(q, 0, q) \mid q \in I\}$.

It follows from the proof of Theorem 12 that for every configuration $p(i)$ of \mathcal{A}_I we have that $Val^{CN}(p(i)) = Val^{NT}(p(i)) = 1$.

Now we give the promised example which demonstrates that the inclusion $OptValOne^{ST} \subseteq ValOne^{ST}$ is proper. Consider the OC-MDP $\hat{\mathcal{A}}$ of the following figure (we draw directly the associated MDP $\mathcal{D}_{\hat{\mathcal{A}}}$):



The control state p is non-deterministic, and the other two control states are stochastic. The probability distributions are always uniform, and the only final control state is s . Now observe that $OptValOne^{ST} = \{s(i) \mid i \in \mathbb{N}_0\}$, while $ValOne^{ST}$ consists of all $p(i)$, $s(i)$, $i \in \mathbb{N}_0$. To see this, let us fix an arbitrarily small $\varepsilon > 0$, and choose some $c \in \mathbb{N}_0$ such that $\frac{1}{2^c} < \frac{\varepsilon}{2}$. We define a MD strategy σ_ε by $\sigma_\varepsilon(p(k)) = p(k+1)$ if $k < c$, and $\sigma_\varepsilon(p(k)) = r(k)$ if $k \geq c$. Now it is easy to check that $\mathcal{P}(ST^{\sigma_\varepsilon}(v)) \geq 1 - \frac{1}{2^c} > 1 - \varepsilon$ for every v of the form $p(i)$, or $s(i)$. On the other hand, there is no strategy σ such that $\mathcal{P}(ST^\sigma(p(i))) = 1$ for any $i \in \mathbb{N}_0$ because every strategy which makes the probability of reaching $s(0)$ from $p(i)$ positive inevitably makes the probability of reaching $r(0)$ positive as well.

Note that the strategy σ_ε from the above example is in fact both MD and FD strategy (see the definition after Lemma 21), i.e. finitely representable by a deterministic finite automaton. This is always the case for strategies approximating the Val^{ST} up to some fixed $\varepsilon > 0$. This is because if some strategy σ satisfies $\mathcal{P}(ST^\sigma(v)) \geq Val^{ST}(v) - \varepsilon/2$ then there is some $n \in \mathbb{N}$ such that the probability of runs from $ST^\sigma(v)$ not longer than n is at least $Val^{ST}(v) - \varepsilon$. On these runs only finitely many configurations appear and thus the choices of σ in these configurations can be kept in a finite memory of a finite automaton. Thus the strategy σ can be replaced by a FD strategy σ' copying the choices of σ until the n -th step. It follows that $\mathcal{P}(ST^{\sigma'}(v)) \geq Val^{ST}(v) - \varepsilon$.

Now we present an exponential-time algorithm which computes an \mathcal{A} -automaton recognizing the set $OptValOne^{ST}$, and we also show that there is a counter-regular strategy σ constructible in exponential time which is optimal in the configurations of $OptValOne^{ST}$. We also give a lower complexity bound and show that deciding the membership to $OptValOne^{ST}$ is **PSPACE**-hard, and the membership to $ValOne^{ST}$ is hard for the Boolean hierarchy over **NP** (note this hierarchy subsumes both **NP** and **coNP**). We did not manage to provide analogous results for $ValOne^{ST}$, and we leave this problem as an open challenge for future work (the above example gives a taste of issues that must be resolved to obtain a solution).

To prove Theorem 15, we need to formulate several auxiliary observations. For every $i \in \mathbb{N}_0$, let

- $Black_i = \{p(i) \in Q \times \mathbb{N}_0 \mid R_i(p) = b\}$
- $White_i = \{p(i) \in Q \times \mathbb{N}_0 \mid R_i(p) = w\}$

Further, let $White = \bigcup_{i \in \mathbb{N}_0} White_i$.

Lemma 28. *There is a MD strategy σ such that for all $0 \leq j < i$ and all $p(i) \in Black_i$ we have that $\mathcal{P}(Reach_{Black_j}^\sigma(p(i))) = 1$ and $\mathcal{P}(Reach_{White}^\sigma(p(i))) = 0$.*

Proof. It is known that for every finitely-branching MDP $\mathcal{D} = (V, \hookrightarrow, (V_N, V_P), Prob)$, every set $T \subseteq V$ of target vertices, and every initial vertex $v \in V$, if there is *some* (i.e., HR) strategy π_v such that $\mathcal{P}(Reach_T^{\pi_v}(v)) = 1$, then there is also a MD strategy σ_v with this property (see, e.g., Theorem 7.2.11 of [23], which applies to more general non-negative bounded total expected reward objectives). The individual

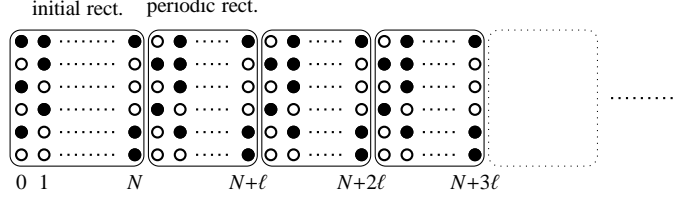


Fig. 2. The structure of coloring R (where $N = 2^{\lfloor \ell \rfloor}$).

MD strategies σ_v can be easily combined into a single MD strategy σ . Since $\mathcal{D}_{\mathcal{A}}^{\rightarrow}$ is finitely-branching, we can apply this generic result and conclude that there is a MD strategy σ such that $\mathcal{P}(ST^{\sigma}(p(i))) = 1$ for every $p(i) \in Q \times \mathbb{N}_0$ where $R(p(i)) = b$. This means that also $\mathcal{P}(Reach_{Q \times \{j\}}^{\sigma}(p(i))) = 1$ for every j such that $0 \leq j < i$. Now suppose that $\mathcal{P}(Reach_{White}^{\sigma}(p(i))) > 0$. Then there is some white configuration $q(j)$ such that $\mathcal{P}(Reach_{\{q(j)\}}^{\sigma}(p(i))) > 0$. Since $q(j)$ is white, we have that $\mathcal{P}(ST^{\sigma}(q(j))) < 1$. Thus, we obtain that $\mathcal{P}(ST^{\sigma}(p(i))) < 1$, which is a contradiction. Since $\mathcal{P}(Reach_{Q \times \{j\}}^{\sigma}(p(i))) = 1$ and $\mathcal{P}(Reach_{White}^{\sigma}(p(i))) = 0$, we have that $\mathcal{P}(Reach_{Black_j}^{\sigma}(p(i))) = 1$. \square

Lemma 14. *There is $1 \leq \ell \leq N$ such that, for every $j \geq N$, the columns $R_j = R_{j+\ell}$.*

Proof. We show that for all $j, k \in \mathbb{N}$ we have that if $R_j = R_k$, then also $R_{j+1} = R_{k+1}$. From this we easily obtain our lemma—since there are at most N different columns, there are $m, n \in \mathbb{N}$ such that $0 \leq m < n \leq N$ and $R_m = R_n$. We put $\ell = n - m$. Obviously, $R_j = R_{j+\ell}$ for every $j \geq m$. Since $m < N$, we are done.

It suffices to prove that for every $i \in \mathbb{N}$, the column R_{i+1} is completely determined by the column R_i in the following sense: For every $q \in Q$ we have that $R_{i+1}(q) = b$ iff there is a strategy σ such that $\mathcal{P}(Reach_{Black_i}^{\sigma}(q(i+1))) = 1$ and $\mathcal{P}(Reach_{White_i}^{\sigma}(q(i+1))) = 0$. Note that the existence of σ does not depend on the exact value of i as long as the column R_i stays the same. Hence, the above claim implies that if $R_j = R_k$, then also $R_{j+1} = R_{k+1}$. It remains to prove this claim. The “ \Rightarrow ” direction follows directly from Lemma 28. For the “ \Leftarrow ” direction, consider a strategy σ such that $\mathcal{P}(Reach_{Black_i}^{\sigma}(q(i+1))) = 1$ and $\mathcal{P}(Reach_{White_i}^{\sigma}(q(i+1))) = 0$. For each $p(i) \in Black_i$ there is a strategy σ_p such that $\mathcal{P}(ST^{\sigma_p}(p(i))) = 1$. Hence, we can construct a strategy π which behaves like σ until some $p(i) \in Black_i$ is reached, and from that point on it behaves like σ_p . Obviously, $\mathcal{P}(ST^{\pi}(q(i+1))) = 1$ as needed. \square

Now we show that the initial and periodic rectangles of the coloring R (given in Figure 2) are computable in exponential time. For this we need to formulate and prove an important observation which establishes a powerful link to the results presented in Section 3. We start by defining a OC-MDP $\mathcal{A}_{R,\ell}$, which encodes the structure obtained by deleting all white points from the periodic rectangle of R . Later, we construct such an automaton also for another coloring B , where some points are gray. Therefore, the definition of $\mathcal{A}_{R,\ell}$ is parametrized by a general coloring which satisfies certain conditions.

Definition 29 (the OC-MDP $\mathcal{A}_{C,\ell}$). *Let $C : Q \times \mathbb{N}_0 \rightarrow \{b, w, g\}$ be a coloring such that $C_N = C_{N+\ell}$ and for every $p(N+i) \in Q \times \mathbb{N}$ where $1 \leq i \leq \ell$ and $C(p(N+i)) \neq w$ we have that*

- (1) *if $p(N+i)$ is probabilistic and $p(N+i) \mapsto q(N+j)$, then $C(q(N+k)) \neq w$, where $k = j \bmod \ell$;*
- (2) *if $p(N+i)$ is non-deterministic, then there is some $p(N+i) \mapsto q(N+j)$ such that $C(q(N+k)) \neq w$, where $k = j \bmod \ell$.*

We define a OC-MDP $\mathcal{A}_{C,\ell}$ where

- the set $Q_{C,\ell}$ of control states of $\mathcal{A}_{C,\ell}$ consists of all $[p, i]$ where $p \in Q$, $1 \leq i \leq \ell$, and $C(p(N+i)) \neq w$. A given control state $[p, i]$ is non-deterministic or probabilistic, depending on whether $p \in Q_N$ or $p \in Q_P$, respectively;
- the set of zero rules consists of all triples $([p, i], 0, [p, i])$, where $[p, i] \in Q_{C,\ell}$;
- the set of positive rules is constructed as follows:
 - for all $(p, c, q) \in \delta^{>0}$ and all $i \in \mathbb{N}$ such that $1 \leq i \leq \ell$, $1 \leq i+c \leq \ell$, and $[p, i], [q, i+c] \in Q_{C,\ell}$, we add a rule $([p, i], 0, [q, i+c])$. If $[p, i]$ is probabilistic, then the probability of the rule $([p, i], 0, [q, i+c])$ is $P^{>0}(p, c, q)$.
 - for all $(p, c, q) \in \delta^{>0}$ and all $i \in \mathbb{N}$ such that $1 \leq i \leq \ell$, $i+c = \ell+1$, and $[p, i], [q, 1] \in Q_{C,\ell}$, we add a rule $([p, i], 1, [q, 1])$. If $[p, i]$ is probabilistic, then the probability of the rule $([p, i], 1, [q, 1])$ is $P^{>0}(p, c, q)$.
 - for all $(p, c, q) \in \delta^{>0}$ and all $i \in \mathbb{N}$ such that $1 \leq i \leq \ell$, $i+c = 0$, and $[p, i], [q, \ell] \in Q_{C,\ell}$, we add a rule $([p, i], -1, [q, \ell])$. If $[p, i]$ is probabilistic, then the probability of the rule $([p, i], -1, [q, \ell])$ is $P^{>0}(p, c, q)$.

Observe that conditions (1) and (2) guarantee that $\mathcal{A}_{C,\ell}$ is indeed an OC-MDP.

Lemma 30. For each configuration $[p, i](j)$ of $\mathcal{A}_{R,\ell}$ we have that $\text{Val}_{\mathcal{D}_{\mathcal{A}_{R,\ell}}^{\rightarrow}}^{NT}([p, i](j)) = 1$.

Proof. Let $[p, i](j)$ be a configuration of $\mathcal{A}_{R,\ell}$. By definition of $\mathcal{A}_{R,\ell}$, we have that $R(p(N+i+j\ell)) = b$. By Lemma 28, there is a MD strategy σ such that $\mathcal{P}(\text{Reach}_{\text{Black}_N}^\sigma(p(i))) = 1$ and $\mathcal{P}(\text{Reach}_{\text{White}}^\sigma(r(m))) = 0$ for every $r(m) \in Q \times \mathbb{N}_0$ where $R(r(m)) = b$. Consider a MD strategy π in $\mathcal{D}_{\mathcal{A}_{R,\ell}}^{\rightarrow}$ defined as follows: for every configuration $[q, k](n)$ of $\mathcal{A}_{R,\ell}$ where $q \in Q_N$ we put $\pi([q, k](n)) = [q', k'](n')$, where

- $\sigma(q(N+k+n\ell)) = q'(t)$,
- $k' = (t - N) \bmod \ell$,
- $n' = (t - N) \div \ell$.

Note that the definition of π is correct, because $R(q'(t)) = b$ and hence the transition $\pi([q, k](n)) = [q', k'](n')$ exists in $\mathcal{D}_{\mathcal{A}_{R,\ell}}^{\rightarrow}$ (realize that if $R(q'(t))$ was white, we would have a contradiction with $\mathcal{P}(\text{Reach}_{\text{White}}^\sigma(q(N+k+n\ell))) = 0$). Since almost all runs of $\mathcal{D}_{\mathcal{A}}^{\rightarrow}(\sigma)$ initiated in $p(N+i+j\ell)$ visit Black_N , we obtain that almost all runs of $\mathcal{D}_{\mathcal{A}_{R,\ell}}^{\rightarrow}(\pi)$ initiated in $[p, i](j)$ visit a configuration of the form $[q, \ell](0)$. This means that $\text{Val}_{\mathcal{D}_{\mathcal{A}_{R,\ell}}^{\rightarrow}}^{NT}([p, i](j)) = 1$. \square

Lemma 31. Let $\mathcal{A} = (Q, \delta^{=0}, \delta^{>0}, (Q_N, Q_P), P^{=0}, P^{>0})$ be a OC-MDP. If $p(i) \mapsto^* q(0)$, then there is a path from $p(i)$ to $q(0)$ in $\mathcal{D}_{\mathcal{A}}^{\rightarrow}$ such that the counter stays bounded by $i+|Q|^2$ along this path.

Proof. For every $j \in \mathbb{N}_0$, we define a relation $\rightsquigarrow_j \subseteq Q \times Q$ inductively as follows:

- $\rightsquigarrow_0 = \{(s, t) \in Q \times Q \mid s(1) \mapsto t(0)\}$
- \rightsquigarrow_{j+1} consists of all $(s, t) \in Q \times Q$ such that one of the following conditions is satisfied:
 - $s \rightsquigarrow_j t$;
 - $s(1) \mapsto r(1)$ for some $r \in Q$ such that $r \rightsquigarrow_j t$;
 - $s(1) \mapsto r(2)$ for some $r \in Q$ such that $r \rightsquigarrow_j u$ and $u \rightsquigarrow_j t$ for some $u \in Q$.

A straightforward induction on j reveals that if $s \rightsquigarrow_j t$, then there is a path from $s(1)$ to $t(0)$ in $\mathcal{D}_{\mathcal{A}}^{\rightarrow}$ along which the counter stays bounded by $j+1$.

Let $\rightsquigarrow = \bigcup_{j \in \mathbb{N}_0} \rightsquigarrow_j$. Observe that $\rightsquigarrow = \rightsquigarrow_{|Q|^2}$. One can easily show that $s \rightsquigarrow t$ iff for every $i \in \mathbb{N}$ there is a path from $s(i)$ to $t(i-1)$ such that the counter is less or equal to $i + |Q|^2$ and greater or equal to i in all configurations except for the last one (the “ \Rightarrow ” direction is proven for every \rightsquigarrow_j by induction on j , and the “ \Leftarrow ” direction is proven by induction on the length of a path from $s(i)$ to $t(i-1)$). From this we get that if there is a path from $s(i)$ to $t(0)$ in $\mathcal{D}_{\mathcal{A}}^{\rightarrow}$ such that the counter stays positive in all configurations except for the last one, then there is a path from $s(i)$ to $t(0)$ along which the counter is bounded by $i + |Q|^2$. Finally, we show that if there is a path from $s(i)$ to $t(0)$ along which the counter becomes zero m times, then there is a path from $s(i)$ to $t(0)$ along which the counter is bounded by $i + |Q|^2$ (this is the result we are aiming at). However, this is easy to prove by induction on m . \square

Lemma 32. *There is a counter-regular strategy σ which is optimal in every configuration of $OptValOne^{ST}$. Further, the underlying \mathcal{A} -automaton and selector function of the strategy σ are computable from the initial and periodic rectangles of the coloring R in time which is exponential in the size of \mathcal{A} .*

Proof. We design a MD strategy π such that

- π is optimal in every configuration of $OptValOne^{ST}$.
- $\pi(p(i)) = \pi(p(i+\ell))$ for all $p \in Q_N$ and $i > |Q|^2 N^2 + N$.
- $\pi(p(i))$ is computable for all $p \in Q_N$ and $i \leq |Q|^2 N^2 + N + \ell$ in time polynomial in N , assuming that the initial and periodic rectangles of R are known.

Obviously, the strategy π can be easily transformed into a counter-regular strategy σ with the required properties.

First, for every $p(i)$ such that $i \leq N$ and $R(p(i)) = b$ we fix a finite path $p(i) \mapsto \dots \mapsto q(0)$ where $q \in F$ and all configurations in the path are black in R . Such a path must exist, and we can further safely assume that the counter stays bounded by $|Q|^2 N^2 + N$ along this path (see Lemma 31) and no configuration appears twice in the path. For all configurations $q(0)$ where $q \in F \cap Q_N$, the strategy π is defined arbitrarily. Now, for each path w fixed above (in any order) we do the following: we identify all non-deterministic configurations $q(j)$ in w for which the strategy π has not yet been defined, and let $\pi(q(j))$ to select the (only) outgoing transition of $q(j)$ that appears in the path w . Let $PathConf$ be the set of all configurations (non-deterministic or probabilistic) that appear in some of the finite paths fixed above.

Now consider again the OC-MDP $\mathcal{A}_{R,\ell}$. According to Theorem 12 and Lemma 30, there is a *CMD* strategy ξ in $\mathcal{D}_{\mathcal{A}_{R,\ell}}^{\rightarrow}$ such that for every configuration $[p, i](j)$ of $\mathcal{A}_{R,\ell}$ we have that $\mathcal{P}(NT^\xi([p, i](j))) = 1$. For every control state $[p, i]$ of $\mathcal{A}_{R,\ell}$ where $p \in Q_N$, let $[p, i](1) \mapsto [q, j](k)$ be the transition selected by $\xi([p, i](1))$. For every $p(N+i+y\ell)$ such that $y \in \mathbb{N}_0$ and $\pi(p(N+i+y\ell))$ has not yet been defined, we let $\pi(p(N+i+y\ell))$ to select the transition $p(N+i+y\ell) \mapsto q(N+j+y\ell+(k-1)\ell)$.

Obviously, we have that $\pi(p(i)) = \pi(p(i+\ell))$ for all $p \in Q_N$ and $i > |Q|^2 N^2 + N$. If the initial and periodic rectangles of R are known, the automaton $\mathcal{A}_{R,\ell}$ is effectively constructible by using Definition 29, and the *CMD* strategy ξ is computable in time polynomial in N by Theorem 12. Hence, $\pi(p(i))$ is computable for all $p \in Q_N$ and $i \leq |Q|^2 N^2 + N + \ell$ in time polynomial in N . To see that π is optimal in every configuration of $OptValOne^{ST}$, realize the following:

- Let $White = \{q(j) \in Q \times \mathbb{N}_0 \mid R(q(j)) = w\}$. Then for every $p(i) \in OptValOne^{ST}$ we have that $\mathcal{P}(Reach_{White}^\pi(p(i))) = 0$.
- Let $fin = \{q(0) \mid q \in F\}$. Then there is a fixed $\varepsilon > 0$ such that for every $p(i) \in PathConf$ we have that $\mathcal{P}(Reach_{fin}^\pi(p(i))) \geq \varepsilon$. This is because for each of the finitely many $p(i) \in PathConf$ there is a finite path from $p(i)$ to fin in $\mathcal{D}_{\mathcal{A}}^{\rightarrow}(\pi)$.

Input: An OC-MDP $\mathcal{A} = (Q, \delta^{=0}, \delta^{>0}, (Q_N, Q_P), P^{=0}, P^{>0})$, a non-empty set $F \subseteq Q$ of final states.

Output: The initial and periodic rectangles of the coloring R .

```

1:  for each  $p(i)$  where  $0 \leq i \leq 2N$  do  $A(p(i)) := w$  done
2:  for each  $\ell$  where  $1 \leq \ell \leq N$  do
3:    for each  $C$  where  $C : Q \rightarrow \{b, w\}$  do
4:      for each  $p(i)$  where  $0 \leq i \leq N + \ell$  do  $B(p(i)) := g$  done
5:      for each  $q \in F$  do  $B(q(0)) := b$  done
6:       $B_N := C; B_{N+\ell} := C$ 
7:      repeat
8:        for each  $p(i)$  where  $0 \leq i \leq N + \ell$  do  $B(p(i)) := \text{check\_color}(p(i))$  done
9:      until  $B$  does not change
10:     if  $B(p(i)) = r$  for some  $p(i)$  then continue with the next  $C$ 
11:     compute the OC-MDP  $\mathcal{A}_{B,\ell}$ 
12:     for each  $p(N+i)$  where  $1 \leq i \leq \ell$  and  $B(p(N+i)) \neq w$  do
13:        $B(p(N+i)) = \text{check\_value}(p(N+i))$ 
14:     done
15:     if  $B(p(i)) = r$  for some  $p(i)$  then continue with the next  $C$ 
16:     repeat
17:       for each  $p(i)$  where  $0 \leq i \leq N$  do  $B(p(i)) := \text{check\_path}(p(i))$  done
18:     until  $B$  does not change
19:     for each  $p(i)$  where  $0 \leq i \leq N$  and  $B(p(i)) = g$  do  $B(p(i)) := b$  done
20:     if  $B(p(i)) = r$  for some  $p(i)$ 
21:       then continue with the next  $C$ 
22:     else transfer all black points of  $B$  to  $A$ 
23:   done
24: done
25: find the least  $\ell$  such that  $A_N = A_{N+\ell}$ 
26: output  $A_0, \dots, A_N$  and  $A_{N+1}, \dots, A_{N+\ell}$ 

```

Fig. 3. An exponential-time algorithm which computes the coloring R

- For each $p(i) \in \text{OptValOne}^{ST} \setminus \text{PathConf}$ we have that $\mathcal{P}(\text{Reach}_{\text{PathConf}}^\pi(p(i))) = 1$. This is because almost all runs in $\mathcal{D}_{\mathcal{A}}^\pi(\pi)$ initiated in $p(i)$ tend to decrease the counter until they reach a configuration of PathConf .

From these three properties, one can conclude that $\mathcal{P}(ST^\pi(p(i))) = 1$ for every $p(i) \in \text{OptValOne}^{ST}$. \square

Theorem 15. *An \mathcal{A} -automaton recognizing the set OptValOne^{ST} is computable in exponential time. Further, there is a counter-regular strategy σ constructible in exponential time which is optimal in every configuration of OptValOne^{ST} .*

Proof. To construct an \mathcal{A} -automaton recognizing the set OptValOne^{ST} , it suffices to compute the initial and periodic rectangles of R . This is achieved by the algorithm given in Fig. 3.

Since the width of the initial rectangle is $N + 1$ and the width of the periodic rectangle is at most N , it suffices to compute the first $2N + 1$ columns of R . For this purpose, we introduce two auxiliary colorings A and B whose domain is restricted to $Q \times \{0, \dots, 2N\}$. The coloring A is just a memory used to accumulate the information about all of the newly discovered black points. The color of all points in A is initially white (line 1) and, as we shall see, each $p(i)$ such that $0 \leq i \leq 2N$ and $R(p(i)) = b$ is eventually recolored to black in A at line 22.

The coloring B is used to discover more and more points that are black in R . This is achieved by trying out all candidates ℓ for the width of the periodic rectangle (line 2) and all candidates C for the column $R_{N+\ell}$

(line 3). For each choice of ℓ and C , the color of all $p(i)$ in B , where $0 \leq i \leq N+\ell$, is first initialized to gray at line 4 (the intuitive meaning of gray is “don’t know”). Then, all $q(0)$ where $q \in F$ are recolored to black at line 5, which is surely correct. Further, the columns $B_{N+\ell}$ and B_N are set to the current candidate C (note $R_{N+\ell} = R_N$). Now, we try to recolor as much points as we can using the function `check_color` (lines 7–9). For a given $p(i)$, where $0 \leq i \leq N+\ell$, the function `check_color` first computes the set of $col(p(i))$ of colors that $p(i)$ should have according to its \mapsto successors and predecessors (we say that $q(j)$ is a \mapsto successor of $r(k)$ if $r(k) \mapsto q(j)$). Formally, $col(p(i))$ is the least set of colors satisfying the following:

- if $p \in Q_P$ and all \mapsto successors of $p(i)$ are black in B , then $b \in col(p(i))$;
- if $p \in Q_P$ and some \mapsto successor of $p(i)$ is white in B , then $w \in col(p(i))$;
- if $p \in Q_N$ and all \mapsto successors of $p(i)$ are white in B , then $w \in col(p(i))$;
- if $p \in Q_N$ and some \mapsto successor of $p(i)$ is black in B , then $b \in col(p(i))$;
- if $q(j) \mapsto p(i)$ where $q \in Q_P$ and $q(j)$ is black in B , then $b \in col(p(i))$;
- if $q(j) \mapsto p(i)$ where $q \in Q_N$ and $q(j)$ is white in B , then $w \in col(p(i))$.

Note that in the case when $i = N+\ell$, we need to know the B color of \mapsto successors and predecessors of $p(i)$ whose counter value can also be $N + \ell + 1$. Here we stipulate that $B(q(N+\ell+1)) = B(q(N+1))$ (note that $R(q(N+\ell+1)) = R(q(N+1))$). Intuitively, `check_color(p(i))` contains the color of $p(i)$ that is “enforced” by the colors of its \mapsto successors and predecessors. If both black and white is enforced, or if $B(p(i))$ is inconsistent with the enforced color, we discovered an inconsistency in the current choice of ℓ and C . Hence, the color which is returned by `check_color(p(i))` is determined as follows:

- if $col(p(i)) = \emptyset$, then `check_color(p(i))` returns $B(p(i))$ (i.e., the current color of $p(i)$ in B);
- if $col(p(i)) = \{c\}$ and $B(p(i)) = g$, then `check_color(p(i))` returns c ;
- if $col(p(i)) = \{c\}$ and $B(p(i)) = c$, then `check_color(p(i))` returns c ;
- in the other cases, `check_color(p(i))` returns r .

Note that the red color is used to mark a consistency error. Also note that each $p(i)$ is recolored at most twice, and so the **repeat-until** loop in lines 7–9 terminates after $O(N)$ iterations, where each iteration invokes the function `check_color` only $O(N)$ times.

After terminating the loop in lines 7–9, the algorithm checks if there is a red $p(i)$ and if it is the case, it rejects the current C and continues with the next candidate (line 10). Otherwise, all points in B are either black, white, or gray, where

- (1) for all $p(i)$ such that $B(p(i)) = g$ we have that `check_color(p(i))` returns g ;
- (2) for all $p(i)$ such that $B(p(i)) \neq g$ we have that if the width of the periodic rectangle of R is ℓ and $R_{N+\ell} = C$ (i.e, the current candidates ℓ and C are the “real” ones), then $B(p(i)) = R(p(i))$. It is easy to show that this claim is an invariant of the **repeat-until** loop in lines 7–9.

Now we need to resolve the color of the remaining gray points. First, we concentrate on the gray points in the columns $B_{N+1}, \dots, B_{N+\ell}$ and check whether they can constitute the periodic rectangle of R after some further recoloring. This is done by checking the condition of Lemma 30. First we construct the OC-MDP $\mathcal{A}_{B,\ell}$ of Definition 29 (line 11). Note that the condition (2) above guarantees that the coloring B satisfies the requirements of Definition 29. For each $p(N+i)$ where $1 \leq i \leq \ell$ and $B(p(N+i)) \neq w$ we recolor $p(N+i)$ to `check_value(B(p(N+i)))` at lines 12–14. Here the function `check_value` does the following: if $B(p(N+i)) = g$, then `check_value(B(p(N+i)))` returns either b or w depending on whether $Val_{\mathcal{D} \rightarrow (\mathcal{A}_{B,\ell})}^{NT}([p, i](j)) = 1$ for all $j \in \mathbb{N}_0$ or not, respectively. If $B(p(N+i)) \neq g$, then `check_value(B(p(N+i)))` returns either b or r , depending on whether $Val_{\mathcal{D} \rightarrow (\mathcal{A}_{B,\ell})}^{NT}([p, i](j)) = 1$ for all $j \in \mathbb{N}_0$ or not, respectively. Note

that $\text{check_value}(B(p(N+i)))$ is computable in time polynomial in the size of N by Theorem 12. Then we check whether some point has been recolored to red, and if it is the case, we continue with the next candidate (line 15). Otherwise, all points in the columns $B_{N+1}, \dots, B_{N+\ell}$ are now black or white. It is important to note that the functions check_color and check_value would not report any inconsistencies in the current B (i.e., if we run the code at lines 7–14 again after line 15, no point would be recolored to red). This follows directly from Remark 27.

It remains to resolve the gray points in the columns B_0, \dots, B_N . Here we use the observation about R formulated in Lemma 32. Let \hat{B} be the (only) coloring satisfying the following conditions:

- $\hat{B}_j = B_j$ for every $0 \leq j \leq N+\ell$;
- $\hat{B}_{N+\ell+i} = \hat{B}_{N+i}$ for every $i \in \mathbb{N}$.

For every $p(i)$ where $0 \leq i \leq N$ and $B(p(i)) \neq w$, we recolor $p(i)$ to $\text{check_path}(p(i))$. The function $\text{check_path}(p(i))$ checks, depending on whether p is probabilistic/non-deterministic, whether for all/some $p(i) \mapsto r(j)$ there is a finite path $r(j) \mapsto \dots \mapsto q(0)$ such that $q \in F$ and all configurations in this path are black or gray in the current \hat{B} . If this is the case, $\text{check_path}(p(i))$ returns the current $B(p(i))$. Otherwise, $\text{check_path}(p(i))$ returns either white or red, depending on whether $B(p(i)) = g$ or $B(p(i)) = b$, respectively. After finishing the loop at lines 16–18, all of the remaining gray points of B_0, \dots, B_N are recolored to black at line 19. Note that the function check_path can be implemented in time polynomial in N by employing, e.g., standard polynomial-time algorithms for the reachability problem in pushdown automata. Then we check whether some point has been recolored to red, and if it is the case, we continue with the next candidate (line 21). Otherwise, all points of $B_0, \dots, B_{N+\ell}$ are black or white. Observe that

- for every $p(i)$ such that $i \leq N$ and $B(p(i)) = b$ there is a finite path $p(i) \mapsto \dots \mapsto q(0)$ where $q \in F$ and all configurations in the path are black in \hat{B} . Further, if $p \in Q_P$ and $p(i) \mapsto r(j)$, then $B(r(j)) = b$.
- there is a *CMD* strategy ξ in $\mathcal{D}_{\mathcal{A}_{B,\ell}}^{\rightarrow}$ such that for every configuration $[p, i](j)$ of $\mathcal{A}_{R,\ell}$ we have that $\mathcal{P}(NT^\xi([p, i](j))) = 1$.

These are *exactly* the ingredients which were needed to construct the strategy π in the proof of Lemma 32. If we apply the same construction to the coloring \hat{B} , we obtain a strategy π_B such that $\mathcal{P}(ST^{\pi_B}(p(i))) = 1$ for every $p(i) \in Q \times \mathbb{N}_0$ where $\hat{B}(p(i)) = b$. This means that all black points in the columns $B_0, \dots, B_{N+\ell}$ can be safely transferred from B to A , which is done at line 22.

After terminating the loop at lines 2–24, the algorithm finds the least ℓ such that $A_N = A_{N+\ell}$, and outputs the rectangles A_0, \dots, A_N and $A_{N+1}, \dots, A_{N+\ell}$. Since the “real” values of ℓ and C are eventually tested as candidates and the algorithms recolors a gray point to a white point only if some condition satisfied by R is violated, all black points of $R_0, \dots, R_{N+\ell}$ are eventually discovered. Since the functions check_color , check_value , and check_path need only polynomial time in the size of N , the whole algorithm is polynomial in the size of N .

After computing the initial and periodic rectangles of R , a counter-regular strategy σ which is optimal for all configurations of OptValOne^{ST} can be constructed by using Lemma 32.

Theorem 16. *Membership in ValOne^{ST} is **BH**-hard. Membership in OptValOne^{ST} is **PSPACE**-hard.*

Proof. We start with proving the **BH**-hardness. Our proof is essentially a variation on a proof by Serre [24] (using a technique that originated in [18] and was later reshaped in [16]) showing that the reachability problem for non-probabilistic 2-player 1-counter games is **DP**-hard. We show that similar arguments work to show **BH**-hardness for OC-MDPs.

First, we show that membership in $ValOne^{ST}$ is **NP**-hard and **coNP**-hard, and then we show how to combine these to get **BH**-hardness.

We start with **NP**-hardness. We reduce from SAT. Suppose we are given a CNF formula $\psi = C_1 \wedge \dots \wedge C_m$, over variables $\{x_1, \dots, x_r\}$. We will encode assignments to the variables of ψ by integers, as follows. Let π_1, \dots, π_r denote the first n prime numbers. Then an integer n corresponds to an assignment that assigns true to x_i if and only if π_i divides n . Note that multiple integers map to the same assignment, but that all assignments are certainly mapped to by some positive integer (e.g., 1 assigns false every variable). It follows from the strong forms of Bertrand's postulate (see, e.g., Theorem 5.8 in [25]) that (as a very conservative bound), for all $r \geq 64$, $\pi_r \leq (2r)^2$. (We can thus of course trivially compute the first r primes π_1, \dots, π_r in time polynomial in r .)

The OC-MDP will have a start state s_0 , which is controlled by the (maximizing) player. The initial configuration is $s_0(1)$ and the player can choose to increment the counter and stay in state s_0 , or to move to state s_1 without changing the counter. Thus, after it has repeatedly incremented the counter up to a "guessed" number $n \geq 0$ which represents an assignment, the game moves to configuration $s_1(n)$.

State s_1 is probabilistic, and it chooses, uniformly at random, one of the clauses C_i , which it claims is not satisfied by the assignment associated with n , and moves to configuration $s'_i(n)$. s'_i is controlled by the maximizing player, and it chooses a literal l_j in C_i , and moves to $s'_{i,l_j}(n)$. Suppose $l_j = x_j$. From this configuration we deterministically decrement the counter, but keep track, using π_j auxiliary states, how many times, mod π_j , we have decremented the counter. Clearly, if we hit the counter value 0 in a state that indicates we have decremented a number of times which is 0 (mod π_j), then the assignment corresponding to n satisfies clause C_i . Similarly, if $l_j = \neg x_j$, we can check that the number of times decremented is $\neq 0$ (mod π_j), in which case again n satisfies clause C_i . Since the random player chose all clauses with equal probability, there is a strategy to terminate in such "accepting" states with probability 1 if there is a satisfying assignment to ψ . Also note that if there is no satisfying assignment to ψ , then there is a fixed $\delta > 0$ such that for every strategy the probability of non-terminating or terminating in a "non-accepting" control state is at least δ . Note that, as it is easy to check using the bound $\pi_r \leq (2r)^2$, the size of the resulting 1C-MDP is polynomial in the size of the formula ψ .

Next, for **coNP**-hardness, suppose we have a CNF formula $\psi = C_1 \wedge \dots \wedge C_m$, over variables $\{x_1, \dots, x_r\}$, and we want to decide unsatisfiability. We do as before, but with some role reversals between non-deterministic and probabilistic control states. Starting in configuration $s_0(1)$ where s_0 is now probabilistic, we randomly either increment the counter or change the state to s_1 (with, say, equal probability). Thus we eventually move to state s_1 with probability 1, and for every positive integer n , with some positive probability we move to (s_1, n) . The state s_1 is controlled (i.e., non-deterministic).

The player's strategy chooses (guesses) a clause C_i which it thinks cannot be satisfied by the assignment n , and moves to configuration $s'_i(n)$, where s'_i is probabilistic. Then the random player picks one of the literals l_j , of clause C_i , uniformly at random (intuitively claiming at least one of them will be satisfied and thus with positive probability we will terminate in a rejecting state), and moves to $s'_{i,l_j}(n)$. We then decrement deterministically as before, except that now when we terminate we accept precisely in those states where we would have not accepted before. Specifically, we accept if "assignment" n did not assign true to literal l_j of clause C_i , which again we can check by keeping track of how many times we decremented mod π_i , upon hitting counter value 0.

Note that under every strategy the probability of termination is 1. Similarly as before, there is a strategy such that the probability of termination in an accepting state is 1 if there is no satisfying assignment to ψ , on the other hand there is some $\delta > 0$ such that terminating in a "non-accepting" state occurs with probability at least δ under every strategy if there is a satisfying assignment to ψ .

Finally, to show **BH**-hardness, consider any statement which is a \wedge - \vee combination of statements of the form “ ψ_i is satisfiable” and “ ψ_j is not-satisfiable”, where ψ_i ’s are Boolean formulas. Deciding whether such statements are true is **BH**-complete. In order to mimic this with a OC-MDP, we do as follows: \vee is mimicked by the controller (i.e., a non-deterministic state) picking one of the disjuncts. \wedge is mimicked by the random player (a probabilistic state) picking one of the conjuncts uniformly at random. When we hit a statement “ ψ_i is (un)satisfiable”, we play the corresponding game. It is easy to check that maximizer has a strategy to terminate in an accepting state with probability 1 if the entire statement is true, and that there is a $\delta > 0$ such that for every strategy termination in an accepting state has probability at most $1 - \delta$ if the entire statement is false.

Note that in all the OC-MDP from the reductions above the sets $OptValOne^{ST}$ and $ValOne^{ST}$ are equal. Thus we have already proved also **BH**-hardness of the membership in both of them. We will now prove, however, that the membership in $OptValOne^{ST}$ is even **PSPACE**-hard.

The proof is by reduction from the emptiness problem for simple alternating finite automata over a one-letter alphabet. A simple alternating finite automaton over a one-letter alphabet (call it AFA for short in the rest of the text) is a tuple (Q, δ, q_0, F) where Q is a finite nonempty set of states, $q_0 \in Q$, $F \subseteq Q$ and δ is a transition function assigning to every state either another state, or the “existential” pair $p \vee q$ of states $p, q \in Q$, or the “universal” pair $p \wedge q$. The automaton is used to recognise sets of words over a one-letter alphabet. Such words can be considered as numbers from \mathbb{N}_0 . The language of the automaton is defined to be the set of exactly those $n \in \mathbb{N}_0$ which are accepted from the state q_0 , written $Acc(q_0, n)$. The semantics of the expression $Acc(q, n)$, meaning accepting a number n from a state q , is defined inductively on n : $Acc(q, 0)$ is true iff $q \in F$. For $n = k + 1$ we have three cases:

- If $\delta(q) = p$ then $Acc(q, k + 1)$ is equivalent to $Acc(p, k)$.
- If $\delta(q) = p_1 \vee p_2$ then $Acc(q, k + 1)$ is true iff at least one of $Acc(p_1, k)$ and $Acc(p_2, k)$ is true.
- If $\delta(q) = p_1 \wedge p_2$ then $Acc(q, k + 1)$ is true iff both $Acc(p_1, k)$ and $Acc(p_2, k)$ are true.

See [17] for more details about AFA. Proposition 4 from [17] states that the problem of deciding whether the language of a given AFA is empty, is **PSPACE**-hard.

We now describe a log-space reduction of the emptiness problem for AFA to the membership in $OptValOne^{ST}$ for OC-MDP. Let (Q, δ, q_0, F) be an AFA. The reduction returns the following OC-MDP: $(Q \cup \{p\}, \delta^{=0}, \delta^{>0}, (Q_N, Q_P), P^{=0}, P^{>0})$ along with the set F of final states and the initial configuration $p(1)$ where

- p is a fresh new state, $p \notin Q$;
- $\delta^{=0} = \{(p, 0, p)\} \cup \{(q, 0, q) \mid q \in Q\}$;
- $\delta^{>0} = \{(p, +1, p), (p, -1, q_0)\} \cup \{(q, -1, r) \mid q, r \in Q, \text{ whenever } r \text{ occurs in } \delta(q)\}$;
- $Q_N = \{p\} \cup \{q \in Q \mid \exists r, s \in Q : \delta(q) = r \vee s\}$, $Q_P = Q \setminus Q_N$;
- the probability assignments always return the uniform distribution.

If n is accepted by the AFA then the following MD strategy σ proves $p(1) \in OptValOne^{ST}$:

- $\sigma(p(n + 1)) = q_0(n)$ and $\sigma(p(k)) = p(k + 1)$ for $k \neq n + 1$,
- $\sigma(q(k)) = r(k - 1)$ for every $q \in Q_N \cap Q$ and $k \in \mathbb{N}$ where r is an arbitrary state occurring in $\delta(q)$ with $Acc(r, k - 1)$ being true, and
- $\sigma(q(k))$ is defined arbitrarily if there is no such r .

On the other hand, if σ ensures almost sure reaching $F \times \{0\}$ from $p(1)$, there must be some n such that $q_0(n)$ is visited on some path from $p(1)$ to $F \times \{0\}$ with positive probability. It can easily be shown that every configuration of the form $q(k)$ visited after $q_0(n)$ satisfies $Acc(q, k)$. In particular $Acc(q_0, n)$ and thus the language of the AFA is not empty. \square

We now show that *qualitative* problems for the special subclass of OC-MDPs given by *solvency games* [1] can be solved in polynomial time. We now recall more formally the definition of solvency games from [1], which was described informally in the introduction. A *solvency game*, is given by a positive integer, n , (n is the initial pot of money belonging to the gambler), and a finite set $\mathcal{A} = \{A_1, \dots, A_k\}$ of *actions* (or “gambles”), each of which is associated with a finite-support probability distribution on the integers. Since for computational purposes we have to be given these distributions as finite input, we assume that the distribution associated with each action A_i , $i = 1, \dots, k$, is encoded by giving a set of pairs $\{(n_{i,1}, p_{i,1}), (n_{i,2}, p_{i,2}), \dots, (n_{i,m_i}, p_{i,m_i})\}$, such that for $j = 1, \dots, m_i$, $n_{i,j} \in \mathbb{Z}$ and $p_{i,j}$ are positive rational probabilities, i.e., $p_{i,j} \in (0, 1]$ and $\sum_{j=1}^{m_i} p_{i,j} = 1$. We assume the integers $n_{i,j}$ and the rational values $p_{i,j}$ are both encoded in the standard way, in *binary* notation.

In a solvency game the player (or *gambler* or *investor*) starts with the initial pot of money, n , and has to repeatedly choose an action (gamble) from the set \mathcal{A} . If at any time the current pot of money is n' , and the gambler then chooses action A_i , then we sample from the finite-support distribution associated with A_i , and the integer, d , resulting from this random sample is added to n' , obtaining the new pot of money $n' + d$. If the pot of money hits 0 or goes below zero, then the gambler loses (goes bankrupt) and the game ends. Otherwise, we repeat the gambling process with the new pot of money $n' + d$. The gambler’s aim is to minimize the probability of ever losing the game, i.e., to minimize the probability of ever going bankrupt. (Note that we do not allow the gambler to simply choose to stop gambling (which would be too easy a way to prevent going bankrupt). Our gamblers are hopelessly addicted! Perhaps then *investor* is more appropriate.)

It should be clear that solvency games constitute a special subclass of OC-MDPs. Namely, the counter in an OC-MDP can be used to 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 additional states and incrementing or decrementing the counter by one at a time. Namely, the OC-MDP will have a “base” *control* state, s , from which is chooses from the set of actions $\{A_1, \dots, A_k\}$. If action A_i , is associated with a probability distribution given by $\{(n_{i,1}, p_{i,1}), (n_{i,2}, p_{i,2}), \dots, (n_{i,m_i}, p_{i,m_i})\}$, we will have $|n_{i,j}|$ additional auxiliary states associated with each such integer $n_{i,j}$ in the support of A_i . After the gambler chooses action A_i , we transition from state s to a new *random* state s_i without changing the counter value. From s_i we move with probability $p_{i,j}$ to a new state $s_{i,j}$, from which we will deterministically (with probability 1) add $n_{i,j}$ to the counter, doing the incrementing or decrementing one at a time, by going through $|n_{i,j}|$ additional states $s_{i,j,1}, \dots, s_{i,j,|n_{i,j}|}$. Finally, after this is done we return to the “base” control state s . It is easy to see that the original solvency game with the objective of minimizing the probability of bankruptcy is equivalent to the resulting OC-MDP, started in state s , with the objective of minimizing the probability of ever reaching counter value 0 (in *any* state). Note that since we assume the integers $n_{i,j}$ are encoded in binary, in principle this reduction yields an OC-MDP that is exponentially larger than the input solvency game. Of course, to make this a polynomial time reduction we can simply assume that the integers $n_{i,j}$ are encoded in unary. Nevertheless, we show that even when the $n_{i,j}$ ’s are encoded in binary, all qualitative problems for solvency games are decidable in polynomial time:

Proposition 17. *Given a solvency game, it is decidable in polynomial time whether the gambler has a strategy to go bankrupt with probability: > 0 , $= 1$, $= 0$, or < 1 .*

Proof. The first three cases (> 0 , $= 1$, $= 0$) are either trivial, or follow fairly easily from what we have established about OC-MDPs, so we do these first. The last case, < 1 , is not easy at all, but follows by using a lovely theorem about non-homogeneous *controlled* random walks by Durrett, Kesten, and Lawler [8].

- > 0 : The gambler has a strategy to go bankrupt with probability > 0 , precisely when there exists an action A_i such that there is a negative number $n_{i,j} < 0$ in its support (i.e., in the support of the corresponding

finite-support distribution on the integers). If such an action A_i exists, then clearly playing action A_i repeatedly yields a non-zero probability of eventually going bankrupt. If no such action exists, then the gambler's wealth never decreases and thus he/she never goes bankrupt, no matter what it does.

= 1: We wish to know whether the gambler has a strategy with which it will go bankrupt with probability 1. (Never mind that the gambler would be stupid to do this.)

Note that, by the reduction to OC-MDPs described above, this case is equivalent to whether in the resulting OC-MDP the controller has a strategy to terminate (i.e., hit counter value 0) in *any* state, with probability 1. Note that this is the *non-selective* termination condition (NT). Thus by Theorem 12, if the supremum probability, over all strategies, of terminating is 1, then there is in fact a *counter-oblivious memoryless* (CMD) optimal strategy, σ , for terminating with probability 1. But note that there is only one controlled state in the OC-MDP (the state s), from which the controller chooses one of the actions A_1, \dots, A_k . Thus, the CMD strategy σ amounts to always choosing the same action, A_i . Translating this strategy back to the solvency game, if the supremum probability of bankruptcy is 1, then there is an optimal action A_i that the gambler should choose repeatedly for ever, which achieves bankruptcy probability = 1.

How do we decide which action does this? This is simple: let the *drift*, $E[A_i]$, associated with an action A_i be the expected change in the counter value if we take action A_i once. This can clearly be computed easily in polynomial time from the description of the probability distribution for A_i .

Note that once we fix an action A_i that we will choose forever, this basically yields a 1-dimensional homogeneous random walk on the integers, starting from a positive integer. It then follows from a basic results in the theory of random walks and sums of i.i.d. random variables (see, e.g., [6] Theorem 8.2.5 and Theorem 8.3.4) that, fixing action A_i , the resulting random walk (starting with a positive wealth) will hit wealth 0 (bankruptcy) with probability 1 if and only if both of the following conditions hold: (1) $E[A_i] \leq 0$ (i.e., the drift is not positive, and (2) A_i has some negative value $n_{i,j} < 0$ in its support.

We can of course check these conditions individually for each action A_i , and we answer yes precisely if some action satisfies these conditions.

= 0: Is there a strategy for the gambler to not go bankrupt with probability 1? Clearly, this is the case if and only if there exists an action A_i which *does not* have a negative number $n_{i,j} < 0$ in its support. It is trivial to check this.

< 1: Finally, we come to the most interesting and difficult case: is there a strategy for the gambler to go bankrupt with probability < 1 , i.e., to not go bankrupt with positive probability?

Note that:

1. If there exists an action A_i which does not have a negative integer $n_{i,j} < 0$ in its support, then playing that action repeatedly suffices to not go bankrupt (in fact to not go bankrupt with probability 1).
2. If there exists an action A_i such that $E[A_i] > 0$ (i.e., whose *drift* is positive), then again by basic facts about random walks and sums of i.i.d. random variables (again, see, e.g., Theorems 8.2.5 and 8.3.4 of [6]), starting with any positive wealth, with positive probability the wealth will never hit 0.

Clearly, both conditions (1.) and (2.) can be checked easily in polynomial time.

Is there any other possible way for the gambler to not go bankrupt with positive probability, perhaps by using some combination of different actions as its strategy? We shall now see that this is not possible. If no action satisfies either of the above two conditions, then there is no strategy at all for the gambler to not go bankrupt with positive probability.

This follows for a lovely (and quite non-trivial to prove) result due to Durrett, Kesten and Lawler [8] about non-homogeneous *controlled* random walks (or, as they put it, about when one can and cannot “*make money from fair games*”). Specifically, Theorem 1 of [8] says the following: suppose a gambler

gets to choose a sequence X_1, X_2, X_3, \dots of *independent* random variables whose range is over the reals, such that the X_i 's, although not necessarily identically distributed, do have the property that they are only *finitely inhomogeneous*, meaning that there exists a finite family of probability distributions $\mathcal{F} = \{F_1, \dots, F_k\}$ over the reals, such that for all $i \in \mathbb{N}$, the distribution of X_i comes from the family \mathcal{F} . Suppose, furthermore, that every distribution in \mathcal{F} has mean 0, i.e., $E[X_i] = 0$, for all i , and has *finite non-zero variance*, i.e., $0 < \text{Var}[X_i] < \infty$, for all i . Let $S_n = \sum_{i=1}^n X_i$, for $n \in \mathbb{N}$. The gambler's strategy can be *adapted*, meaning its choice of distribution for X_i can depend on the outcomes from X_1, \dots, X_{i-1} . Theorem 1 of [8] says that as long as these conditions hold, the sequence of random variables S_n is *recurrent*, meaning there is some $0 < L < \infty$ such that $\text{Prob}(S_n \in [-L, L] \text{ i.o.}) = 1$, or in other words, such that the probability that $S_n \in [-L, L]$ infinitely often (i.e., for infinitely many n) is 1.¹⁰ Note that this also means that for an fixed value $D < 0$, with probability 1 the sequence S_n will eventually hit a value $\leq D$. (This is because it will have infinitely many "shots" at hitting a value $\leq D$ from a starting point inside the interval $[-L, L]$, and each such shot has a positive probability which is bounded away from 0 by a positive $\epsilon > 0$. This later fact holds because there are only finitely many distributions to choose from, and each distribution is non-trivial because it has non-zero variance.)

Let us see now why this implies that the *only* conditions under which the gambler has a strategy not to go bankrupt with positive probability are when either one of conditions (1.) or (2.) above hold.

Consider the set of actions A_1, \dots, A_k . Suppose neither condition (1.) nor (2.) holds for any of these actions. Thus, each action A_i has some negative integer $n_{i,j} < 0$ in its support, and furthermore no action A_i has positive drift, i.e., for all actions A_i , $E[A_i] \leq 0$.

Let us first assume that all actions have drift 0, i.e., for all i , $E[A_i] = 0$. In this case, since each action has a negative integer in its support, clearly $\text{Var}[A_i] > 0$. Furthermore, for every i the distribution of A_i has only finite support, clearly $\text{Var}[A_i] < \infty$. Thus we are in exactly the situation of Theorem 1 of [8], and consequently we know that regardless of what wealth D we start with, with probability 1 the wealth will eventually hit a value ≤ 0 .

What if there are some actions A_i for which $E[A_i] < 0$? Well, intuitively, this can only favor the probability of bankruptcy. More formally, we can do as follows: for every action A_i with $E[A_i] < 0$, obtain a new random variable A'_i from A_i by letting $A'_i = A_i - E[A_i]$. Clearly, $E[A'_i] = 0$. Furthermore, $0 < \text{Var}[A'_i] < \infty$, because the same holds for A_i . Thus, for these revised random variables, again, the condition holds that starting from any positive wealth the gambler eventually goes bankrupt with probability 1, regardless of the strategy. But sums of these revised random variables are always just rightward translations of sums of the original set of random variables. So if we go bankrupt with probability 1 with the revised random variables, then we would also go bankrupt with probability 1 with the original random variables. This completes the proof.

Thus checking cases (1.) and (2.) for each action yields a correct polynomial time algorithm for determining whether there is a strategy for the gambler to not go bankrupt with positive probability.

□

C Why bounding the counter can yield bad approximations

As discussed in the introduction, here is a simple example for why cutting off the counter at a finite value, even for a purely stochastic QBD (equivalently, a probabilistic one-counter automaton) can in general radically alter its behavior. Consider a 2-state QBD which in state 1, with probability $p = 1/2^n$ goes to state 2,

¹⁰ Incidentally, in [8] they also note that without the condition that $\text{Var}[X_i] < \infty$, there are simple examples where $S_n \rightarrow \infty$ almost surely. In other words, without such conditions on higher moments, one can indeed *make money from fair games*.

and with probability $1 - p$ stays in state 1, in both cases incrementing the counter, and in state 2 stays in state 2 with probability 1 and decrements the counter. We are interested in the probability of termination starting at state 1, with counter value 1. By cutting off the counter at a value $N \in 2^{o(n)}$ the termination probability goes down to ϵ arbitrarily close to 0, for large enough n . Although we used small probabilities $1/2^n$ in this example, the same thing can easily be achieved using a QBD with $O(n)$ states and only the probability $1/2$ on transitions.