The Verification of Probabilistic Systems Under Memoryless Partial-Information Policies is Hard[*]

Luca de Alfaro

Department of Electrical Engineering and Computer Sciences,
University of California at Berkeley.
Email: dealfaro@eecs.berkeley.edu

Abstract

Several models of probabilistic systems comprise both probabilistic and nondeterministic choice. In such models, the resolution of nondeterministic choices is mediated by the concept of *policies* (sometimes called *adversaries*). A policy is a criterion for choosing among nondeterministic alternatives on the basis of the past sequence of states of the system. By fixing the resolution of nondeterministic choice, a policy reduces the system to an ordinary stochastic system, thus making it possible to reason about the probability of events of interest.

A *partial information* policy is a policy that can observe only a portion of the system state, and that must base its choices on finite sequences of such partial observations. We argue that in order to obtain accurate estimates of the worst-case performance of a probabilistic system, it would often be desirable to consider partial-information policies. However, we show that even when considering memoryless partial-information policies, the problem of deciding whether the system can stay forever with positive probability in a given subset of states becomes NP-complete. As a consequence, many verification problems that can be solved in polynomial time under perfect-information policies, such as the model-checking of pCTL or the computation of the worst-case long-run average outcome of tasks, become NP-hard under memoryless partial-information policies. On the positive side, we show that the worst-case long-run average outcome of tasks under under memoryless partial-information policies can be computed by solving a nonlinear programming problem, opening the way to the use of numerical approximation algorithms.

1 Introduction

In several models of probabilistic systems, probabilistic and nondeterministic choice coexist. While probabilistic choice provides a statistical characterization of the system behavior, nondeterminism is used to model concurrency [Var85, PZ86, SL94], and lack of knowledge of transition probabilities [Seg95, dA97] and transition rates [dA98b]. In such a model, the probability of events depends on the way the nondeterministic choices are resolved during the behavior of the system. To assign a probability to the events, it is

[*]This research was supported in part by the NSF CAREER award CCR-9501708, by the DARPA (NASA Ames) grant NAG2-1214, by the DARPA (Wright-Patterson AFB) grant F33615-98-C-3614, by the ARO MURI grant DAAH-04-96-1-0341, and by the Gigascale Silicon Research Center.

customary to use the notion of *policy* [Bel57], closely related to the *schedulers* of [Var85] and the *adversaries* of [SL94]. Whenever the choice among nondeterministic alternatives arises, a policy dictates the probability of choosing each alternative, possibly as a function of the past sequence of states visited by the system. Hence, once the policy is specified, the nondeterminism present in the system is resolved, and the system is thus reduced to a purely probabilistic system. In the statement of verification problems, nondeterminism is usually assigned a *demonic* role: a property is considered to hold iff it holds under any possible resolution of nondeterministic choice, or equivalently, under any policy. Perfect-information and partial-information policies correspond to demons with different observation powers. A *perfect-information* policy is a policy that can observe the complete description of the system state, and that can select among the nondeterministic alternatives on the basis of the finite sequence of states traversed by the system. In contrast, a *partial-information* policy can only observe part of the system state, and it must select among the alternatives on the basis of finite sequences of such incomplete observations.

To understand why partial-information policies can lead to more accurate estimates of the worst-case system performance, consider the following example. Consider a telecommunication network that routes phone calls between nodes, and consider two users u_1 and u_2, attached to two nodes of the network. When user u_1 tries to call user u_2, he is either connected to user u_2, or he receives a busy signal, indicating that there are no connections available to route the call. We intend to model the system in order to study the long-run average fraction of successful calls. To simplify the example, we assume that we have enough statistical information about the network to model the number of connections available between any pair of nodes as a purely probabilistic process, without any nondeterminism. On the other hand, we do not have precise information on when our particular user u_1 wishes to call u_2, so that we model the choice to place a call as a nondeterministic choice. From the point of u_1, a state s of the system consists of four components $s = (s[c], s[u_1], s[r], s[n])$, where:

- $s[c]$ is a portion of the state visible to everyone (e.g., the current time of the day);
- $s[u_1] \in \{\mathit{idle}, \mathit{trying}, \mathit{connected}\}$ describes the state of u_1;
- $s[n] \in \{0, \ldots, N\}$ is the number of available connections for calls between u_1 and u_2; if $s[n] = 0$, then no call from u_1 to u_2 can take place.
- $s[t]$ is a portion of the state visible only to the network (e.g., the state of other communication links and routing tables).

If $s[u_1] = \mathit{idle}$, there is a nondeterministic choice between staying at *idle* (i.e., not placing a call), or going to *trying* (i.e., dialing the number and waiting for the connection to be established). If $s[u_1] = \mathit{trying}$ and $s[n] > 0$, then the connection is established, and u_1 proceeds to *connected*; if $s[u_1] = \mathit{trying}$ and $s[n] = 0$, user u_1 gets a busy signal, and returns to *idle*. If $s[u_1] = \mathit{connected}$, user u_1 can either remain in this state, or hang up and proceed to *idle*. In Section 3 we present the formal model of a system similar to the one described above.

If we model the communication system as indicated above, and study the system under perfect-information policies, we obtain that in the worst case the long-run fraction of successful calls of is 0 — independently of how many free connections there are on average between u_1 and u_2! In fact, at states where $s[u_1] = \mathit{idle}$ the choice of whether to stay at *idle* or try to place a call (going to *connected*) is nondeterministic. In such a state, a perfect-information policy can look at the value of $s[n]$ before deciding whether to place a call. Hence, a worst-case perfect-information policy will place a call only from states where $s[n] = 0$, i.e., where all the connections between u_1 and u_2 are busy. While the value 0 is indeed a lower bound for the long-run average fraction of successful calls from u_1 to u_2, this answer takes an unrealistic, and overly pessimistic, view of the system. In fact, the use of nondeterminism to model the decision of u_1 to place a call is intended to model an unknown dependency between the frequency with which u_1 places calls, and global information such as the time of the day. It is unrealistic to assume that u_1 can base its decision to place a call on the number of free connections, since such information would not be available to u_1 in a real telecommunication system. In order to obtain a more realistic worst-case analysis, we need to consider partial-information policies, which can base their decision of whether to place a call on the state of u_1 and on global information in $s[c]$, but not on information that is internal to the network, such as the number of free connections between u_1 and u_2.

The telecommunication example also suggests why the need for partial-information policies is more felt in the analysis of probabilistic systems than in the analysis of purely nondeterministic ones. In a purely nondeterministic system, we are generally interested in the possibility of events, rather than in their frequency. Hence, all *finite* sequence of events, however rare they might be, are taken into account for establishing a property. In purely nondeterministic systems, the concept of *fairness* is normally used instead of partial information to guard against an infinite number of unfortunate coincidences, such as trying to place a call always only when no connection is free. In the above example, we can rule out such behaviors by adding a fairness condition to the system, requiring that the choice between placing a call and staying in *idle* should be (strongly) fair at all states. Even though fairness and partial information are not equivalent, fairness is preferred because it is amenable to simpler verification methods. However, fairness is not a substitute for partial information in the study of the *frequency* of events. For example, the above fairness condition on placing calls does not ensure that the *frequency* of placing calls is not influenced by the number of free connections. In fact, even with this fairness condition, the worst-case long-run average fraction of calls that are successful is arbitrarily close to 0: the fairness condition is still satisfied if a fraction of the calls smaller than 1, but arbitrarily close to 1, is placed from states where no connection is free. We note that fairness in probabilistic systems can be used as a surrogate for partial information when the specification languages cannot refer to the frequency of events, as is the case for the logic pCTL [BK98, dA99].

Our model for systems with both probabilistic and nondeterministic choice is that of *Markov decision processes* [Bel57, Der70], which are closely related to several models proposed in the literature [Var85, PZ86, SL94]. We consider the *confinement problem,* consisting in deciding whether there is a policy in a given class of policies that enables us to stay forever in a specified subset of states with probability greater than 0. While the confinement problem is solvable in polynomial-time for general policies, from [Rei84] we have that the confinement problem is EXPTIME-complete for partial-information policies. We show that the confinement problem is NP-complete even if we restrict our attention to *memoryless* and *limit-memoryless* partial-information policies. Limit-memoryless partial-information are policies whose state-action frequencies converges to that of a memoryless partial-information policy. These results imply that the model-checking problem for pCTL specifications, and the problem of computing the worst-case long-run average outcome of tasks [dA98a], which can be solved in polynomial time under perfect-information policies, are EXPTIME-hard under partial-information policies, and NP-hard under memoryless and limit-memoryless partial-information policies. On the positive side, we show that the worst-case long-run average outcome of tasks under memoryless and limit-memoryless partial-information policies can be computed by solving a nonlinear optimization problem.

The paper is organized as follows. In Section 2 we describe Markov decision processes and partial-information policies, and we define the *confinement* problem. In Section 3 we present the machinery for defining the long-run average outcome of tasks, and we describe a simple telecommunication example that helps to motivate the consideration of partial-information policies. In Section 4 we present lower-bound results on the complexity of pCTL model checking and computation of long-run average outcomes under partial information. Section 5 presents the optimization problem that enables the computation of the worst-case long-run average outcome of tasks under memoryless partial-information policies, and Section 6 contains some concluding comments.

2 Markov Decision Processes and Partial-Information Policies

Our model for probabilistic systems is a *Markov decision process* (MDP). An MDP is a generalization of a Markov chain in which a set of possible actions is associated with each state. To each state-action pair corresponds a probability distribution on the states, which is used to select the successor state [Der70]. Markov decision processes are closely related to the *probabilistic automata* of [Rab63], the *concurrent Markov chains* of [Var85], and the *simple probabilistic automata* of [SL94, Seg95]. Given a countable set C we denote by $\mathcal{D}(C)$ the set of probability distributions over C, i.e. the set of functions $f : C \mapsto [0,1]$ such that $\sum_{x \in C} f(x) = 1$. An MDP $\mathcal{P} = (S, \mathit{Acts}, A, p)$ consists of the following components:

- A set S of states.
- A set Acts of actions.
- A function $A : S \mapsto 2^{\mathit{Acts}}$, which associates with each $s \in S$ a finite set $A(s) \subseteq \mathit{Acts}$ of actions available at s.
- A function $p : S \times \mathit{Acts} \mapsto \mathcal{D}(S)$, which associates with each $s, t \in S$ and $a \in A(s)$ the probability $p(s,a)(t)$ of a transition from s to t when action a is selected.

We measure the complexity of the algorithms as a function of the *size* of the MDP $\mathcal{P}$, defines as $\sum_{s \in S} |A(s)|$. A *path* of an MDP is an infinite sequence $s_0, a_0, s_1, a_1, \ldots$ of alternating states and actions, such that $s_i \in S$, $a_i \in A(s_i)$ and $p(s_i, a_i)(s_{i+1}) > 0$ for all $i \geq 0$. For $i \geq 0$, the sequence is constructed by iterating a two-phase selection process. First, an action $a_i \in A(s_i)$ is selected nondeterministically; second, the successor state s_{i+1} is chosen according to the probability distribution $p(s_i, a)$. Given a path $s_0, a_0, s_1, a_1, \ldots$ and $k \geq 0$, we denote by X_k, Y_k its generic k-th state s_k and its generic k-th action a_k, respectively. For $n \geq 0$, we call a finite portion $s_0, a_0, s_1, \ldots, s_n$ of path a *finite path prefix.*

Let S^+ be the set of non-empty finite sequences of states. A (perfect-information) *policy* π is a mapping $\pi : S^+ \mapsto \mathcal{D}(\mathit{Acts})$, which associates with each sequence of states

$\bar{s} : s_0, s_1, \ldots, s_n \in S^+$ and each $a \in A(s_n)$ the probability $\pi(\bar{s})(a)$ of choosing a after following the sequence of states $\bar{s}$. We require that $\pi(\bar{s})(a) > 0$ implies $a \in A(s_n)$: a policy can choose only among the actions that are available at the state where the choice is made. According to this definition, policies are randomized, differently from the *schedulers* of [Var85, PZ86], which are deterministic. We indicate with Π the set of all policies. We say that a policy π is *memoryless* if $\pi(\bar{s}, s) = \pi(\bar{t}, s)$ for all $\bar{s}, \bar{t} \in S^*$ and all $s \in S$.

To define partial-information policies, we define *partial-information relations.* A partial-information relation for an MDP $\mathcal{P} = (S, \mathit{Acts}, A, p)$ is an equivalence relation $\sim \subseteq S \times S$ such that for all $s \sim t$, we have $A(s) = A(t)$. If two states are related by $\sim$, then the states cannot be distinguished by a partial-information policy. The condition on $\sim$ ensures that if two states cannot be distinguished by the policy, then the policy can choose among the same actions at the two states. Given two sequences of states $\bar{s} : s_0, \ldots, s_n$ and $\bar{t} : t_0, \ldots, t_m$, with $m, n \geq 0$, we write $\bar{s} \sim \bar{t}$ iff $m = n$ and $s_i \sim t_i$ for all $1 \leq i \leq n$. Given an MDP $\mathcal{P}$ and a partial-information relation $\sim$ for $\mathcal{P}$, we say that a policy π is *partial-information* iff $\pi(\bar{s}) = \pi(\bar{t})$ for all $\bar{s}, \bar{t} \in S^+$ such that $\bar{s} \sim \bar{t}$. If the relation $\sim$ has been fixed, we denote by *PIPol* the set of partial-information policies with respect to $\sim$.

Once a policy π has been selected, the Markov decision process is reduced to a purely probabilistic process, and it becomes possible to define the probabilities of events. In particular, the probability of following a finite path prefix $s_0, a_0, s_1, a_1, \ldots, s_n$ under policy $\pi \in \Pi$ is given by

$$\mathrm{Pr}^{\pi}_{s_0}(X_0 = s_0 \wedge Y_0 = a_0 \wedge \cdots \wedge X_n = s_n) \;=\; \prod_{i=0}^{n-1} p(s_i, a_i)(s_{i+1})\, \pi(s_0, \ldots, s_i)(a_i) \;.$$

To extend this probability measure to subsets of infinite paths, for every state $s \in S$ we denote by Θ_s the set of (infinite) paths having s as initial state. Given two paths (or path prefixes) θ_1 and θ_2, we denote by $\theta_1 \preceq \theta_2$ the fact that θ_1 is a prefix of θ_2. Following the classical definition of [KSK66], we let $\mathcal{B}_s \subseteq 2^{\Theta_s}$ be the σ-algebra of *measurable* subsets of Θ_s, defined as the smallest algebra that contains all the *cylinder sets* $\{\theta \in \Theta_s \mid \sigma \preceq \theta\}$, for σ that ranges over all finite path prefixes, and that is closed under complementation and countable unions (and hence also countable intersections). The elements of $\mathcal{B}_s$ are called *events,* and they are the measurable sets of paths to which we will associate a probability. For $\mathcal{A} \in \bigcup_{s \in S} \mathcal{B}_s$, we write $\mathrm{Pr}^{\pi}_{s}(\mathcal{A})$ to denote the probability of event $\mathcal{A} \cap \mathcal{B}_s$ starting from the initial state $s \in S$ under policy π, and we write $\mathrm{E}^{\pi}_{s}\{f\}$ to denote the expectation of the random function $f : \Theta_s \mapsto \mathbb{R}$ from initial state s under policy π.

The Confinement Problem

Given an MDP $\mathcal{P} = (S, \mathit{Acts}, A, p)$, a subset $U \subseteq S$, a state $s \in S$, and a class of policies $\mathcal{C}$, the *confinement problem* consists in determining whether there is a policy $\pi \in \mathcal{C}$ such that

$$\mathrm{Pr}^{\pi}_{s}\Big(\forall k \geq 0 \,.\, X_k \in U\Big) > 0 \;. \tag{1}$$

It is known that for $\mathcal{C} = \Pi$ the confinement problem can be solved in polynomial-time with efficient graph algorithms. We shall study the complexity of this problem for partial-information policies. We note that the confinement problem is at the heart of several algorithms for the model-checking of pCTL* specifications [CY95, BdA95, BK98]; hence, the complexity of the confinement problem directly affects the complexities of these model-checking problems.

3 Long-Run Average Outcome

The long-run average properties considered in [dA98a, dA98b] refer to the average outcome of a task, which is repeated infinitely often during the behavior of the system. During a task, a certain amount of *outcome* is accrued, indicating for instance the time required to complete the task, or the successful or unsuccessful completion of the task. In our telecommunication example, a task consists in trying to place a call; the outcome accrued is 1 if the call succeeds, and 0 if no connection is available. The long-run average outcome of this task is equal to the long-run average fraction of successful calls. Given an MDP $\mathcal{P} = (S, \mathit{Acts}, A, p)$, we specify tasks and outcomes using two labelings r and w. The labeling $w : S \times \mathit{Acts} \mapsto \{0, 1\}$ associates with each $s \in S$ and each $a \in A(s)$ the value 1 if taking action a at s signals the completion of a task, and value 0 otherwise. The labeling $r : S \times \mathit{Acts} \mapsto \mathbb{R}$ associates an outcome to each state-action pair. We say that a policy π is *proper* from $s \in S$ such that

$$\lim_{n\to\infty} \mathrm{E}^{\eta}_{s}\Big\{\sum_{k=0}^{n-1} w(X_k, Y_k)\Big\} = \infty \ ,$$

indicating that the system performs an infinite expected number of experiments from s. We denote by $\mathit{PropPol}(s) \subseteq \Pi$ the set of proper policies from s. Given $s \in S$ and a proper policy $\pi \in \mathit{PropPol}(s)$, we define the *long-run average outcome* v^{π}_{s} of π from s by

$$v^{\pi}_{s} = \liminf_{n\to\infty} \frac{\mathrm{E}^{\eta}_{s}\Big\{\sum_{k=0}^{n-1} r(X_k, Y_k)\Big\}}{\mathrm{E}^{\eta}_{s}\Big\{\sum_{k=0}^{n-1} w(X_k, Y_k)\Big\}} \ .$$

Given a class of policies $\mathcal{C}$, let $\mathit{PropS}(\mathcal{C}) = \{s \in S \mid \mathit{PropPol}(s) \cap \mathcal{C} \neq \emptyset\}$ be the set of states with at least one proper policy belonging to the class. The *minimum long-run average outcome* problem consists in computing

$$v^{-}_{\mathcal{C}} = \min_{s\in \mathit{PropS}(\mathcal{C})} \Big\{ \inf_{\pi\in\mathcal{C}\cap \mathit{PropPol}(s)} v^{\pi}_{s} \Big\} \ ,$$

assuming that $\mathit{PropS}(\mathcal{C}) \neq \emptyset$. If $\mathcal{C} = \Pi$, this problem can be solved in polynomial time by a reduction to linear programming [dA97, dA98a].

A Simple Telecommunication Example

The following example presents a discrete-time model of a telecommunication system similar to the one discussed in the introduction. The example illustrates the use of nondeterminism for the representation of approximate knowledge of transition probabilities. Consider a telecommunication network, in which there is a total number $n > 0$ of connections available, and there is a distinguished user u_1 that tries intermittently to place calls. We model this system by the MDP $\mathcal{P} = (S, \mathit{Acts}, A, p)$, where $S = \{\mathit{idle}, \mathit{trying}, \mathit{connected}\} \times \{0, \ldots, n\} \times \{0, 1\}$. In a state $\langle x, k, i\rangle \in S$, x is the state of the user u_1, k is the number of busy connections, and i specifies whether it is u_1's turn

$(i = 0)$, or the network's turn $(i = 1)$ of updating the state. The actions are $\mathit{Acts} = \{a, b\}$, and we have $A(s) = \{a, b\}$ for every $s \in S$.

The number of free connections performs a random walk between 0 and n. From state $\langle x, k, 1\rangle$, under either action a or b, we update the state as follows:

- If $2 \leq k < n$, then we go with probability 1/2 to $\langle x, k-1, 0\rangle$ and with probability 1/2 to $\langle x, k+1, 0\rangle$.
- If $k = 1$ and $x \neq \mathit{connected}$, then we go with probability 1/2 to $\langle x, k-1, 0\rangle$ and with probability 1/2 to $\langle x, k+1, 0\rangle$.
- If $k = 1$ and $x = \mathit{connected}$, then we go to $\langle x, k+1, 0\rangle$.
- If $k = 0$, then we go to $\langle x, k+1, 0\rangle$.
- If $k = n$, then we go to $\langle x, k-1, 0\rangle$.

From state $\langle \mathit{idle}, k, 0\rangle$, if the action a is chosen we proceed to state $\langle \mathit{idle}, k, 1\rangle$; if action b is chosen, we proceed to state $\langle \mathit{trying}, k, 0\rangle$. From state $\langle \mathit{trying}, k, 0\rangle$, under both actions a and b we proceed to state $\langle \mathit{connected}, k+1, 1\rangle$ if $k < n$, and to state $\langle \mathit{idle}, k, 1\rangle$ if $k = n$. Finally, from state $\langle \mathit{connected}, k, 0\rangle$ under both actions a and b we proceed to state $\langle \mathit{idle}, k-1, 1\rangle$: for simplicity, we consider only unit-duration phone calls.

To measure the long-run average fraction of successful calls, we define the labels r and w as follows. We let $w(\langle \mathit{trying}, k, 0\rangle, \xi) = 1$ for all $0 \leq k \leq n$ and all $\xi \in \{a, b\}$, so that w counts the number of attempted calls. We let $r(\langle \mathit{trying}, n, 0\rangle, \xi) = 0$ and $r(\langle \mathit{trying}, k, 0\rangle, \xi) = 1$ for all $0 \leq k < n$ and all $\xi \in \{a, b\}$, so that r counts the number of successful calls. It is easy to check that v^s_π is equal to the fraction of successful calls from the initial state s under policy π. The worst-case value of this fraction under perfect-information policies is $v^-_\Pi = 0$. This worst-case value arises when the user u_1 chooses action b whenever there are no free connections, and action a otherwise. This is clearly an unrealistic worst-case value. A better estimate can be obtained by introducing a partial-information relation $\sim$ defined by $\langle i, k_1, j\rangle \sim \langle i, k_2, j\rangle$ for all $i \in \{\mathit{idle}, \mathit{trying}, \mathit{connected}\}$, all $0 \leq k_1, k_2 \leq n$, and all $j = 0, 1$. This partial-information relation prevents the user u_1 from selecting actions a and b on the basis of the number of free connections. The worst-case fraction of successful calls under partial-information policies v^-_{PIPol} provides a more realistic estimate of the performance of the system.

We note that introducing a partial-visibility relation is equivalent to assuming that there are no factors external to the model that can influence the policies differently at states related by the partial-visibility relation. While u_1 most likely cannot base his decision of calling on the number of free connections, there might be external factors that make it more likely for u_1 to call when more connections are busy. For example, in countries where soccer is popular, more people place telephone calls during the mid-game intervals than during the game proper. If such external factors are not accounted for, then the worst-case long-run fraction of successful calls computed under partial-information policies is an optimistic estimate of the true worst-case long-run average fraction. Hence, adding partial-information restrictions to the policies should be done on the basis of a careful examination of the model.

In alternative to using partial information, we can increase the accuracy of the worst-case estimates of the fraction of successful calls by reducing the role of nondeterminism and providing more probabilistic information about the user's behavior. Specifically, suppose that we know that the probability that the user will place a call when idle is between 0.1 and 0.2. To represent this range, we modify the above model as follows. From state $\langle \mathit{idle}, k, 1\rangle$ action a (resp. b) leads to $\langle \mathit{idle}, k, 1\rangle$ with probability 0.9 (resp. 0.8), and to $\langle \mathit{trying}, k, 0\rangle$ with probability 0.1 (resp. 0.2). In this model, v_{Π}^{-} may provide a realistic value for the fraction of successful calls, and the resolution of the remaining nondeterminism under perfect information can account for correlations of events not described by the model, such as the above-mentioned soccer-game phenomenon.

4 Complexity of Partial-Information Confinement

In this section, we present the complexity results for the confinement problem under partial-information policies. By reasoning as in [Rei84], it can be shown that the confinement problem is EXPTIME complete for partial-information policies. Moreover, we show that the confinement problem is NP-complete for memoryless and limit-memoryless partial-information policies.

4.1 General Partial-Information Policies

The following theorem states our result for general partial-information policies.

Theorem 1 *The confinement problem is EXPTIME-complete for the class of partial-information policies.*

Proof. The fact that the problem is in EXPTIME follows from the subset construction of [Rei84]. The lower bound follows by reasoning as in [Rei84] for "blindfold games", repeating infinitely many times the simulation of the nondeterministic Turing machine. ∎

Corollary 1 *The problems of pCTL model checking and of the computation of the minimum long-run average outcome are EXPTIME-hard for incomplete-information policies.*

Proof. The result about pCTL model checking follows directly from Theorem 1 by considering a property of the form $\Box U$, where by abuse of notation we denote by U both a subset of states, and a predicate defining such subset. The result about the minimum long-run average outcome follows from Theorem 1 by considering an MDP in which the set U, once left, cannot be re-entered, together with a function w identically equal to 1, and a function r defined by $r(s, a) = 1$ if $s \in U$ and $r(s, a) = 0$ if $s \notin U$, for all states s and actions a. ∎

4.2 Memoryless and Limit-Memoryless Partial-Information Policies

A *memoryless partial-information policy* is a policy that is both memoryless and partial information. We denote by Π_{MP} the class of memoryless partial-information policies. To define limit-memoryless partial-information policies, for all $s \in S$ and $a \in A(s)$ we denote by

$$N_{s,a}^{n} = \sum_{k=0}^{n-1} \delta(X_k = s \wedge Y_k = a)$$

the random variable indicating the number of times that the state-action pair s, a appears in the first n steps of a path. A *frequency-stable* policy is a policy π such that the limit

$$x_t^\pi(s,a) = \lim_{n\to\infty} \frac{1}{n} \mathrm{E}_t^\pi\{N_{s,a}^n\} \qquad (2)$$

exists for all $t, s \in S$ and $a \in A(s)$. The quantity $x_t^\pi(s,a)$ is the frequency of state-action pair s, a from the initial state t under policy π. A policy π is *limit-memoryless partial information* if it is frequency stable, and if for all states $s, t, u \in S$ with $t \sim u$, one of the following condition holds:

1. either $x_s^\pi(t,a) = 0$ for all $a \in A(t)$;
2. or $x_s^\pi(u,a) = 0$ for all $a \in A(u)$;
3. or, for all $a \in A(t)$,

$$\frac{x_s^\pi(t,a)}{\sum_{b\in A(t)} x_s^\pi(t,b)} = \frac{x_s^\pi(u,a)}{\sum_{b\in A(u)} x_s^\pi(u,b)} \ . \qquad (3)$$

Together, these conditions state that each action is chosen with the same relative frequency at s and at t, unless one of s or t has 0 frequencies for all the actions. We denote by Π_{LMP} the class of limit-memoryless partial-information policies. We note that a memoryless partial-information policy is also a limit-memoryless partial-information policy. Intuitively, a limit-memoryless partial-information policy is a policy that can initially behave in an arbitrary way, but that on the long run gives rise to state-action frequencies that correspond to those of a memoryless partial-information policy.

Theorem 2 *The confinement problem for the classes of memoryless and limit-memoryless policies is NP-complete.*

Proof. To see that the problems are in NP, note that it suffices to guess a deterministic memoryless partial-information policy, and check that it satisfies (1). The proof of NP-hardness is by a reduction from the SAT problem. Consider an instance of SAT problem defined over a finite set $Y = \{y_1, \ldots, y_k\}$ of variables. Let $L = Y \cup \{\bar{y} \mid y \in Y\}$ be the set of literals, and let $c_1, \ldots, c_n \subseteq L$ be the clauses composing the problem. The SAT problem consists in checking whether the propositional formula $\bigwedge_{i=1}^n \bigvee_{l\in c_i} l$ is satisfiable. From this instance of SAT problem, we construct an MDP $\mathcal{P} = (S, \mathit{Acts}, A, p)$ and a partial information relation $\sim$ as follows. The state space is

$$S = \{s_0, s_1\} \cup \{1, \ldots, k+1\} \times \{1, \ldots, n\} \times \{0, 1\} \ .$$

We let $U = S \setminus \{s_0\}$, and we take s_1 as the initial state. A state of the form $\langle m, i, j\rangle$ refers to the occurrence of variable y_m in clause c_i (where y_{k+1} is a dummy variable). The component j of the state keeps track of whether clause c_i has already been satisfies by the variable assignment ($j = 1$) or not ($j = 0$). The set of actions is $\mathit{Acts} = \{a, b\}$. We have $A(\langle m, i, j\rangle) = \{a, b\}$ for all $1 \le m \le k+1$, all $1 \le i \le n$, and all $j = 0, 1$. Choosing action a (resp. b) at state $\langle m, i, j\rangle$ corresponds to choosing the truth value *true* (resp. *false*) for y_m in clause c_i. We let $A(s_0) = A(s_1) = \{a\}$. The transitions are as follows. We let

$p(s_0, a)(s_0) = 1$, so that s_0 is absorbing, and we let $p(s_1, a)(\langle 1, i, 0\rangle) = 1/n$, for $1 \leq i \leq n$. For all $1 \leq i \leq n$ and $\xi \in \{a, b\}$, we let

$$p(\langle k+1, i, 1\rangle, \xi)(s_1) = 1 \qquad p(\langle k+1, i, 0\rangle, \xi)(s_0) = 1$$

so that if the clause i has been satisfied, we go back to s_1, and we proceed to s_0 otherwise. For $1 \leq m \leq k$, $1 \leq i \leq n$, and $j = 0, 1$, the transitions from the other states are defined as follows:

- From $\langle m, i, 1\rangle$, both a and b lead deterministically to $\langle m+1, i, 1\rangle$.
- From $\langle m, i, 0\rangle$, we have three cases:
  - If $y_m \in c_i$, then a leads deterministically to $\langle m+1, i, 1\rangle$ and b to $\langle m+1, i, 0\rangle$.
  - If $\bar{y}_m \in c_i$, then a leads deterministically to $\langle m+1, i, 0\rangle$ and b to $\langle m+1, i, 1\rangle$.
  - If $y_m \not\in c_i$ and $\bar{y}_m \not\in c_i$, then both a and b lead deterministically to $\langle m+1, i, 0\rangle$.

Finally, the partial information relation is defined by $\langle m, i_1, j_1\rangle \sim \langle m, i_2, j_2\rangle$ for all $1 \leq m \leq k+1$, all $1 \leq i_1, i_2 \leq n$, and all $j_1, j_2 \in \{0, 1\}$. The states s_0 and s_1 are equivalent only to themselves.

The idea of the construction is as follows. From s_1, the process proceeds uniformly at random to a state of the form $\langle 1, i, 0\rangle$, for $1 \leq i \leq n$. The following k choices between actions a and b correspond to the choice of a truth assignment for variables $y_1, \ldots, y_k$. If the truth assignment satisfies the clause c_i, the process goes to $\langle k+1, i, 1\rangle$; otherwise, it goes to $\langle k+1, i, 0\rangle$. From $\langle k+1, i, 1\rangle$, the process goes back to s_1, and it selects randomly another clause to test. From $\langle k+1, i, 0\rangle$, which indicates that clause c_i has not been satisfies, we go to $s_0 \not\in U$, which indicates failure. Since the policy does not know which one of the clauses $c_1, \ldots, c_n$ is being tested, the only way for the policy to stay in U forever with probability greater than 0 is to select a truth assignment that satisfies simultaneously all the clauses. In the other direction, from a truth assignment that satisfies all clauses we can immediately derive a memoryless partial-information policy that never leaves U. Hence, the confinement problem has an affirmative answer iff the SAT instance is satisfiable. We note that this proof also shows the NP-completeness of the confinement problem for general partial-information policies. ∎

Corollary 2 *The problems of pCTL model checking and of the computation of the minimum long-run average outcome are NP-hard for memoryless or limit-memoryless incomplete-information policies.*

The proof of this corollary is similar to the proof of Corollary 1.

5 Verification under Memoryless Partial-Information Policies

In this section, we show how the minimum long-run average outcome under memoryless or limit-memoryless partial-information policies corresponds to the solution of a nonlinear optimization problem. Even though solving this problem is NP-hard, as shown in the previous section, we can use techniques for the approximate solution of nonlinear optimization problems to obtain upper bounds for the minimum long-run average outcome. These upper bounds can be used in the analysis of the performance of the system. An overview of techniques for the solution of nonlinear optimization problems can be found in [Ber95a].

Restricting the attention to memoryless or limit-memoryless partial-information policies, rather than considering general ones, is often not a drawback. In fact, it is possible to model as part of the state of the system any information about the past history of the system that can influence the resolution of nondeterminism. Additionally, the goal of partial information is to limit the power of the demonic resolution of nondeterminism; often, the further limitation of lack of memory is quite natural in a performance-evaluation setting. In particular, if nondeterminism is used to model unknown values for transition probabilities, rather than concurrency, then it is appropriate to resolve nondeterminism in a memoryless fashion. Finally, we recall that under perfect information, there are always worst-case policies for pCTL and long-run average outcome specifications that are memoryless. Hence, the consideration of memoryless policies to compute the worst case under partial information is a fairly natural extension.

Consider an MDP $\mathcal{P} = (S, \mathit{Acts}, A, p)$, together with two labelings $w : S \times \mathit{Acts} \mapsto \{0,1\}$ and $r : S \times \mathit{Acts} \mapsto \mathbb{R}$. Assume also that $\mathit{PropS}(\Pi) \neq \emptyset$ and $\mathit{PropS}(\Pi_{LMP}) \neq \emptyset$. The minimum long-run average outcome under perfect-information policies v^-_Π can be computed by solving the following linear-programming problem [Ber95b, dA98a].

LP Problem P1. *Set of variables:* $\{\lambda\} \cup \{h_s \mid s \in S\}$.
Maximize λ *subject to:*

$$h_s \leq r(s,a) - \lambda w(s,a) + \sum_{t \in S} p(s,a)(t)\, h_t \qquad \text{for all } s \in S \text{ and } a \in A(s) \qquad \blacksquare$$

To compute $v^-_{\Pi_{LMP}}$ and $v^-_{\Pi_{MP}}$, we take the dual of the above linear-programming problem, and we add a (nonlinear) constraint encoding (3). The resulting nonlinear-programming problem is given below.

Optimization Problem P2. *Set of variables:* $\{x_{s,a} \mid s \in S \wedge a \in A(s)\}$.

Minimize $\sum_{s \in S} \sum_{a \in A(s)} x_{s,a} R(s,a)$ *subject to:*

$$x_{s,a} \geq 0 \qquad \text{for all } s \in S \text{ and } a \in A(s) \tag{4}$$

$$\sum_{s \in S} \sum_{a \in A(s)} x_{s,a} p(s,a)(t) = \sum_{b \in A(t)} x_{t,b} \qquad \text{for all } t \in S \tag{5}$$

$$\sum_{s \in S} \sum_{a \in A(s)} x_{s,a} w(s,a) = 1 \tag{6}$$

$$x_{s,a} \sum_{b \in A(t)} x_{t,b} = x_{t,a} \sum_{b \in A(s)} x_{s,b} \qquad \text{for all } s,t \in S \text{ with } s \sim t \text{ and all } a \in A(s). \tag{7}$$

$\blacksquare$

The meaning of this optimization problem is as follows. For all $s \in S$ and $a \in A(s)$, the variables $x_{s,a}$ are proportional to the state-action frequencies defined in Section 4.2. Equation (4) simply states that all variables are positive. Equation (5) is a flow constraint, requiring that for every state, the frequency of entering the state is equal to the frequency of leaving it. Equation (6) is a normalization constraint, that renormalizes the state-action frequencies so that the (adjusted) frequency of completing a task is 1. Equation (7) encodes directly the constraint (3). The goal of the optimization problem is to minimize the outcome received per unit of frequency. Because of (6), this is equivalent to minimizing the outcome per task. The following theorem states that the above optimization problem computes the desired quantity.

Theorem 3 *The solution of the nonlinear programming problem P2 is equal to the minimum long-run average outcome under memoryless or limit-memoryless partial-information policies.*

6 Conclusions

In this paper, we argued that the accurate estimation of worst-case performance properties of systems that include probabilistic and nondeterministic choice requires the consideration of partial-information policies. On the other hand, we showed that even for memoryless partial-information policies, the problem of computing the worst-case long-run average outcome is NP-hard. We then presented a non-linear optimization problem whose solution enables the computation of performance indices of a system under partial-information policies.

These results point to some future directions for the modeling and analysis of long-run average properties of probabilistic systems (such as performance). One direction consists in using nondeterminism in the model sparingly, remembering that it will be resolved under perfect information, and relying on a manual inspection of the worst-case scenarios to determine their plausibility. A second direction of research consists in identifying a concept that captures some of the relevant features of partial-information policies, while leading to polynomial-time verification algorithms. A third direction consists in studying the system under memoryless or limit-memoryless partial-information policies, and in devising algorithms that, while NP-complete in the worst case, exhibit good average-case complexity for typical system models.

References

[ASB+95] A. Aziz, V. Singhal, F. Balarin, R.K. Brayton, and A.L. Sangiovanni-Vincentelli. It usually works: The temporal logic of stochastic systems. In *Computer Aided Verification*, volume 939 of *Lect. Notes in Comp. Sci.* Springer-Verlag, 1995.

[BdA95] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Found. of Software Tech. and Theor. Comp. Sci.*, volume 1026 of *Lect. Notes in Comp. Sci.*, pages 499–513. Springer-Verlag, 1995.

[Bel57] R.E. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

[Ber95a] D.P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1995.

[Ber95b] D.P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, 1995. Volumes I and II.

[BK98] C. Baier and M. Kwiatkowska. Model checking for a probabilistic branching time logic with fairness. *Distr. Comp.*, 11, May 1998.

[CY95] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42(4):857–907, 1995.

[dA97] L. de Alfaro. *Formal Verification of Probabilistic Systems.* PhD thesis, Stanford University, 1997. Technical Report STAN-CS-TR-98-1601.

[dA98a] L. de Alfaro. How to specify and verify the long-run average behavior of probabilistic systems. In *Proc. 13th IEEE Symp. Logic in Comp. Sci.*, pages 454–465, 1998.

[dA98b] L. de Alfaro. Stochastic transition systems. In *CONCUR'98: Concurrency Theory. 9th Int. Conf.*, Lect. Notes in Comp. Sci., pages 423–438. Springer-Verlag, 1998.

[dA99] L. de Alfaro. From fairness to chance. *Electronic Notes in Theoretical Computer Science*, 1999. Accepted for publication.

[Der70] C. Derman. *Finite State Markovian Decision Processes.* Academic Press, 1970.

[KSK66] J.G. Kemeny, J.L. Snell, and A.W. Knapp. *Denumerable Markov Chains.* D. Van Nostrand Company, 1966.

[PZ86] A. Pnueli and L. Zuck. Probabilistic verification by tableaux. In *Proc. First IEEE Symp. Logic in Comp. Sci.*, pages 322–331, 1986.

[Rab63] M.O. Rabin. Probabilistic automata. *Information and Computation*, 6:230–245, 1963.

[Rei84] J.H. Reif. The compexity of two-player games of incomplete information. *Journal of Computer and System Sciences*, 29:274–301, 1984.

[Seg95] R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems.* PhD thesis, MIT, 1995. Technical Report MIT/LCS/TR-676.

[SL94] R. Segala and N.A. Lynch. Probabilistic simulations for probabilistic processes. In *CONCUR'94: Concurrency Theory. 5th Int. Conf.*, volume 836 of *Lect. Notes in Comp. Sci.*, pages 481–496. Springer-Verlag, 1994.

[Var85] M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state systems. In *Proc. 26th IEEE Symp. Found. of Comp. Sci.*, pages 327–338, 1985.