Journal of Computer and System Sciences 84 (2017) 144–170

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Trading performance for stability in Markov decision processes

Tomáš Brázdil [a,*], Krishnendu Chatterjee [b], Vojtěch Forejt [c], Antonín Kučera [a]

[a] Faculty of Informatics, Masaryk University, Czechia
[b] IST Austria, Austria
[c] Department of Computer Science, University of Oxford, United Kingdom

ARTICLE INFO

Article history:
Received 9 February 2016
Received in revised form 16 July 2016
Accepted 21 September 2016
Available online 19 October 2016

Keywords:
Markov decision processes
Mean payoff
Stability
Stochastic systems
Controller synthesis

ABSTRACT

We study controller synthesis problems for finite-state Markov decision processes, where the objective is to optimize the expected mean-payoff performance and stability (also known as variability in the literature). We argue that the basic notion of expressing the stability using the statistical variance of the mean payoff is sometimes insufficient, and propose an alternative definition. We show that a strategy ensuring both the expected mean payoff and the variance below given bounds requires randomization and memory, under both the above definitions. We then show that the problem of finding such a strategy can be expressed as a set of constraints.

1. Introduction

Markov decision processes (MDPs) are a standard model for stochastic dynamic optimization. Roughly speaking, an MDP consists of a finite set of states, where in each state, one of the finitely many actions can be chosen by a controller. For every action, there is a fixed probability distribution over the states. The execution begins in some initial state where the controller selects an outgoing action, and the system evolves into another state according to the distribution associated with the chosen action. Then, another action is chosen by the controller, and so on. A *strategy* is a recipe for choosing actions. In general, a strategy may depend on the execution history (i.e., actions may be chosen differently when revisiting the same state) and the choice of actions can be randomized (i.e., the strategy specifies a probability distribution over the available actions). Fixing a strategy for the controller makes the behaviour of a given MDP fully probabilistic and determines the usual probability space over its *runs*, i.e., infinite sequences of states and actions.

A fundamental concept of performance and dependability analysis based on MDP models is *mean payoff*. Let us assume that every action is assigned some rational *reward*, which corresponds to some costs (or gains) caused by the action. The mean payoff of a given run is then defined as the long-run average reward per executed action, i.e., the limit of partial averages computed for longer and longer prefixes of a given run. For every strategy σ, the overall performance (or throughput) of the system controlled by σ then corresponds to the expected value of mean payoff, i.e., the *expected mean payoff*. It is well known (see, e.g., [23]) that optimal strategies for minimizing/maximizing the expected mean payoff are positional (i.e., deterministic and independent of execution history), and can be computed in polynomial time. However, the quality of ser-

* Corresponding author.
E-mail addresses: xbrazdil@fi.muni.cz (T. Brázdil), krish.chat@gmail.com (K. Chatterjee), vojfor@cs.ox.ac.uk (V. Forejt), tony@fi.muni.cz (A. Kučera).

http://dx.doi.org/10.1016/j.jcss.2016.09.009

vices provided by a given system often depends not only on its overall performance, but also on its *stability* (sometimes also called *variability*). For example, an optimal controller for a live video streaming system may achieve the expected throughput of approximately 2 MBits/sec. That is, if a user connects to the server many times, he gets 2 Mbits/sec connection *on average*. If an acceptable video quality requires at least 1.8 Mbits/sec, the user is also interested in the likelihood that he gets at least 1.8 Mbits/sec. That is, he requires a certain level of *overall stability* in service quality, which can be measured by the *variance* of the mean payoff, called *global variance* in this paper. The basic computational question is *"given rationals u and v, is there a strategy that achieves the expected mean payoff u (or better) and variance v (or better)?"*. Since the expected mean payoff can be "traded" for smaller global variance, we are also interested in approximating the associated *Pareto curve* consisting of all points (u, v) such that (1) there is a strategy achieving the expected mean payoff u and global variance v; and (2) no strategy can improve u or v without worsening the other parameter.

The global variance says how much the actual mean payoff of a run tends to deviate from the expected mean payoff. However, it does not say *anything* about the stability of individual runs. To see this, consider again the video streaming system example, where we now assume that although the connection is guaranteed to be fast on average, the amount of data delivered per second may change substantially along the executed run for example due to a faulty network infrastructure. For simplicity, let us suppose that performing one action in the underlying MDP model takes one second, and the reward assigned to a given action corresponds to the amount of transferred data. The above scenario can be modelled by saying that 6 Mbits are downloaded every third action, and 0 Mbits are downloaded in other time frames. Then the user gets 2 Mbits/sec connection almost surely, but since the individual runs are apparently "unstable", he may still see a lot of stuttering in the video stream. As an appropriate measure for the stability of individual runs, we propose *local variance*, which is defined as the long-run average of $(r_i(\omega) - mp(\omega))^2$, where $r_i(\omega)$ is the reward of the i-th action executed in a run ω and $mp(\omega)$ is the mean payoff of ω. Hence, local variance says how much the rewards of the actions executed along a given run deviate from the mean payoff of the run on average. For example, if the mean payoff of a run is 2 Mbits/sec and all of the executed actions deliver 2 Mbits, then the run is "absolutely smooth" and its local variance is zero. The level of "local stability" of the whole system (under a given strategy) then corresponds to the *expected local variance*. The basic algorithmic problem for local variance is similar to the one for global variance, i.e., *"given rationals u and v, is there a strategy that achieves the expected mean payoff u (or better) and the expected local variance v (or better)?"*. We are also interested in the underlying Pareto curve.

Observe that the global variance and the expected local variance capture different and to a large extent *independent* forms of systems' (in)stability. Even if the global variance is small, the expected local variance may be large, and vice versa.

1.1. The results

Our results are as follows:

1. *(Global variance).* The global variance problem was considered before in [26], but only under the restriction of memoryless strategies. We first show that in general, randomized memoryless strategies are not sufficient for Pareto optimal points for global variance (Example 1). We then establish that 2-memory strategies are sufficient, and that the problem of existence of a strategy can be reduced to the problem of finding a solution of a set of non-linear constraints. We show that the basic algorithmic problem for global variance is in PSPACE, and the approximate version can be solved in pseudo-polynomial time.
2. *(Local variance).* The local variance problem comes with new conceptual challenges. For example, for unichain MDPs, deterministic memoryless strategies are sufficient for global variance, whereas we show (Example 2) that even for unichain MDPs both randomization and memory are required for local variance. We establish that 3-memory strategies are sufficient for Pareto optimality for local variance, and again give a set of non-linear constraints describing the existence of a strategy. We show that the basic algorithmic problem (and hence also the approximate version) is in NP.
3. *(Zero variance).* Finally, we consider the problem where the variance is optimized to zero (as opposed to a given non-negative number in the general case). In this case, we present polynomial-time algorithms to compute the optimal mean-payoff that can be ensured with zero variance (if zero variance can be ensured) for both the cases. The polynomial-time algorithms for zero variance for mean-payoff objectives is in sharp contrast to the NP-hardness for cumulative reward MDPs [19].

To prove the above results, one has to overcome various obstacles. For example, although at multiple places we build on the techniques of [13] and [2] which allow us to deal with maximal end components (sometimes called strongly communicating sets) of an MDP separately, we often need to extend these techniques. Unlike the works [13] and [2] which study multiple "independent" objectives, in the case of the global variance any change of value in the expected mean payoff implies a change of value of the variance. Also, since we do not impose any restrictions on the structure of the strategies, we cannot even assume that the limits defining the mean payoff and the respective variances exist; this becomes most apparent in the case of the local variance, where we need to rely on delicate techniques of selecting runs from which the limits can be extracted. Another complication is that while most of the work on multi-objective controller synthesis for MDPs deals with linear objective functions, our objective functions are inherently quadratic due to the definition of variance. Finally,