Institute of Science and Technology

Unifying Two Views on Multiple Mean-Payoff Objectives in Markov Decision Processes

Krishnendu Chatterjee and Zuzana Komarkova and Jan Kretinsky

Technical Report No. IST-2015-318-v1+1
Deposited at 12 Jan 2015 13:48
http://repository.ist.ac.at/318/1/main.pdf

IST Austria (Institute of Science and Technology Austria)
Am Campus 1
A-3400 Klosterneuburg, Austria

Unifying Two Views on Multiple Mean-Payoff Objectives in Markov Decision Processes*

Krishnendu Chatterjee[1], Zuzana Komárková[2], and Jan Křetínský[1]

[1] IST Austria
[2] Masaryk University, Brno, Czech Republic

Abstract. We consider Markov decision processes (MDPs) with multiple limit-average (or mean-payoff) objectives. There have been two different views: (i) the expectation semantics, where the goal is to optimize the expected mean-payoff objective, and (ii) the satisfaction semantics, where the goal is to maximize the probability of runs such that the mean-payoff value stays above a given vector. We consider the problem where the goal is to optimize the expectation under the constraint that the satisfaction semantics is ensured, and thus consider a generalization that unifies the existing semantics. Our problem captures the notion of optimization with respect to strategies that are risk-averse (i.e., ensures certain probabilistic guarantee). Our main results are algorithms for the decision problem which are always polynomial in the size of the MDP. We also show that an approximation of the Pareto-curve can be computed in time polynomial in the size of the MDP, and the approximation factor, but exponential in the number of dimensions. Finally, we present a complete characterization of the strategy complexity (in terms of memory bounds and randomization) required to solve our problem.

1 Introduction

MDPs and mean-payoff objectives. The standard models for dynamic stochastic systems with both nondeterministic and probabilistic behaviors are Markov decision processes (MDPs) [How60,Put94,FV97]. An MDP consists of a finite state space, and in every state a controller can choose among several actions (the nondeterministic choices), and given the current state and the chosen action the system evolves stochastically according to a probabilistic transition function. Every transition in an MDP is associated with a reward (or cost), and the basic problem is to obtain a strategy (or policy) that resolves the choice of actions in order to optimize the rewards obtained over the run of the system. An objective is a function that given a sequence of rewards over the run of the system combines them to a single value. A classical and one of the most well-studied objective in context of MDPs is the *limit-average (or mean-payoff)* objective that assigns to every run the average of the rewards over the run.

* Preliminary report

Single vs multiple objectives. MDPs with single mean-payoff objectives have been widely studied in the literature (see, e.g., [Put94,FV97]), with many applications ranging from computational biology, to analysis of security protocols, randomized algorithms, or robot planning, to name a few [BK08,KNP02,DEKM98,KGFP09]. In verification of probabilistic systems, MDPs are widely used such as for concurrent probabilistic systems [CY95,Var85], probabilistic systems operating in open environments [Seg95,dA97], and applied in diverse domains [BK08,KNP02]. However, in several application domains, there is not a single optimization goal, but multiple, potentially dependent and conflicting goals. For example, in designing a computer system, the goal is to maximize average performance while minimizing average power consumption, or in an inventory management system, the goal is to optimize several potentially dependent costs for maintaining each kind of product. These motivate the study of MDPs with multiple mean-payoff objectives, that has also been applied in several problems such as dynamic power management [FKP12].

Two views. There exist two views in the study of MDPs with mean-payoff objectives [BBC$^+$14]. The traditional and classical view is the *expectation* semantics, where the goal is to maximize (or minimize) the expectation of the mean-payoff objective. There are numerous applications of MDPs with expectation semantics, such as in inventory control, planning, and performance evaluation [Put94,FV97]. The alternative semantics is called the *satisfaction* semantics, which given a mean-payoff value threshold sat and a probability threshold pr asks for a strategy to ensure that the probability of runs such that the mean-payoff value is at least sat is at least pr. In case of n reward functions, there are two possible interpretations. Let $\boldsymbol{sat}$ and $\boldsymbol{pr}$ be two vectors of thresholds of dimension k, and $0 \leq pr \leq 1$ be a single threshold. The first interpretation (namely, the *conjunction interpretation*) requires the satisfaction semantics in each dimension $1 \leq i \leq n$ with thresholds $\boldsymbol{sat}_i$ and $\boldsymbol{pr}_i$, respectively, (where $\boldsymbol{v}_i$ is the i-th component of vector $\boldsymbol{v}$). The second interpretation (namely, the *joint interpretation*) requires the satisfaction semantics with probabilistic threshold value pr and the desired set of runs are runs where the mean-payoff value vector is at least $\boldsymbol{sat}$. The distinction of the two views (expectation vs satisfaction) and their applicability in analysis of problems related to stochastic reactive systems has been discussed in details in [BBC$^+$14]. While the joint interpretation of satisfaction has already been introduced [BBC$^+$14], here we consider also the conjunctive interpretation, which was not considered in [BBC$^+$14].

Our problem. In this work we consider a new problem that unifies the two different semantics. Intuitively, the problem we consider asks to *optimize* the expectation while *ensuring* the satisfaction semantics. Formally, consider an MDP with n reward functions, a probability threshold vector $\boldsymbol{pr}$ (or threshold pr for joint interpretation), and a mean-payoff value threshold vector $\boldsymbol{sat}$. We consider the set of *satisfaction* strategies that ensure the satisfaction semantics. Then the optimization of the expectation is considered with respect to the satisfaction strategies. Note that if $\boldsymbol{pr}$ is $\mathbf{0}$, then the satisfaction strategies is the set of all strategies and we obtain the traditional expectation semantics as a spe-

cial case. We also consider important special cases of our problem, depending on whether there is a single reward (mono-reward) or multiple rewards (multi-reward), and whether the probability threshold $\boldsymbol{pr} = \mathbf{1}$ (qualitative criteria) or the general case (quantitative criteria). Specifically, we consider four cases: (1) *Mono-qual:* we have a single reward function and qualitative satisfaction semantics; (2) *Mono-quant:* we have a single reward function and quantitative satisfaction semantics; (3) *Multi-qual:* we have multiple reward functions and qualitative satisfaction semantics; (4) *Multi-quant:* we have multiple reward functions and quantitative satisfaction semantics. Note that for multi-qual case the two interpretations (conjunction and joint) of the satisfaction semantics coincide, whereas in the multi-quan problem (which is the most general problem) we consider both the conjunction as well as the joint interpretations.

Motivation. The motivation to study the problem we consider is twofold. First, it presents a unifying approach that combines the two existing semantics for MDPs. Second, it allows us to consider the problem of optimization along with *risk aversion*. A risk-averse strategy must ensure certain probabilistic guarantee on the payoff function. The notion of risk aversion is captured by the satisfaction semantics, and thus the problem we consider captures the notion of optimization under risk-averse strategies that provide probabilistic guarantee. The notion of *strong risk-aversion* where the probability is treated as an adversary is considered in [BFRR14]; whereas we consider probabilistic (both qualitative and quantitative) guarantee for risk aversion. We now illustrate our problem with several examples.

Illustrative examples:

- For simple risk aversion, consider a single reward function modelling investment. Positive reward stands for profit, negative for loss. We aim at maximizing the expected long-run average while guaranteeing that it is positive with at least 95%. This is an instance of *mono-quant* with $pr = 0.95$, $sat = 0$.
- For more dimensions, consider the example [Put94, Problems 6.1, 8.17]. A vendor assigns to customers either a low or a high rank. Further, there is a decision vendor makes each year to send them a catalogue or not. Depending on their rank and on receiving a catalogue, they spend different amounts and also can change their rank. The aim is to maximize the expected profit provided the catalogue is not sent too often. This is an instance of *multi-qual*. Further, we can extend this example to only require that the catalogue is not sent too often with 95% probability, but 5% best customers may still receive many catalogues (instance of *multi-quant*).
- The following is again an instance of *multi-quant*. A gratis service for dowloading is offered as well as a premium one. For each we model the throughput as rewards r_1, r_2. Expected throughput $1Mbps$ is guaranteed from the gratis service. For a premium service, not only we have a higher expectation of $10Mbps$, but also 95% of the connections are guaranteed to run on at least $5Mbps$. In order to keep this guarantee, we may need to temporarily hire resources from a cloud, whose cost is modelled as a reward r_3. We now want to maximize the expectation of $p_2 \cdot r_2 - p_3 \cdot r_3$ where p_2 is the price per Mb

at which the premium service is sold and p_3 is the price at which additional servers can be hired.

The basic computational questions. In MDPs with multiple mean-payoff objectives, different strategies may produce incomparable solutions. Thus, there is no "best" solution in general. Informally, the set of *achievable solutions* is the set of all vectors $\boldsymbol{v}$ such that there is a strategy that ensures the satisfaction semantics and that the expected mean-payoff value vector under the strategy is at least $\boldsymbol{v}$. The "trade-offs" among the goals represented by the individual mean-payoff objectives are formally captured by the *Pareto curve*, which consists of all maximal tuples (with respect to componentwise ordering) that are not strictly dominated by any achievable solution. Pareto optimality has been studied in cooperative game theory [Owe95] and in multi-criterion optimization and decision making in both economics and engineering [Kos88,YC03,SCK04].

We study the following fundamental questions related to the properties of strategies and algorithmic aspects in MDPs:

- *Strategy complexity:* What type of strategies is sufficient (and necessary) for achievable solutions?
- *Algorithmic complexity:* What is the complexity to decide whether a given vector represents an achievable solution, and if the answer is yes, then compute a witness strategy?
- *Pareto-curve computation:* Is it possible to compute an approximation of the Pareto curve?

Our contributions. We provide comprehensive answers to the above questions. The main highlights of our contributions are:

- *Strategy complexity.* It is known that for both expectation and satisfaction semantics with single reward deterministic memoryless[1] strategies are sufficient. We show this carries over in the *mono-qual* case only. In contrast, for *mono-quant* both randomization and memory is necessary, and we establish that the memory size is dependent on the MDP; the result also applies to the expectation problem of [BBC$^+$14], where no lower bound was given. However, we also show that only a restricted form of randomization (so-called deterministic update) is necessary even for *multi-quant*, thus improving the result for the satisfaction problem of [BBC$^+$14]. A complete picture of the strategy complexity, and improvement over previous results is given in Table 1 and Remark 2 on p. 24.
- *Algorithmic complexity.* We present algorithms for deciding whether a given vector is an achievable solution, and all our algorithms are polynomial in the size of the MDP. Moreover, they are polynomial even in the number of dimensions, except for *multi-quant* with conjunction interpretation where it is exponential.

[1] A strategy is memoryless if it is independent of the history, but depends only on the current state. A strategy that is not deterministic is called randomized

– *Pareto-curve computation.* We show that in all cases with multiple rewards an ϵ-approximation of the Pareto curve can be achieved in time polynomial in the size of the MDP, exponential in the number of dimensions, and polynomial in $\frac{1}{\epsilon}$, for $\epsilon > 0$.

Technical contributions. In the study of MDPs (with single or multiple rewards), the solution approach is often by characterizing the solution as a set of linear constraints. Similar to the previous works [CMH06,EKVY08,FKN$^+$11,BBC$^+$14] we also obtain our results by showing that the set of achievable solutions can be represented by a set of linear constraints, and from the linear constraints witness strategies for achievable solutions can be constructed. However, while previous work on the satisfaction semantics [BBC$^+$14,RRS14] reduces the problem to calling linear programming for each maximal end-component and another linear program putting partial results together, we unify the solution approaches for expectation and satisfaction and provide one complete linear program for the whole problem. This in turn allows us to optimize the expectation *while* guaranteeing satisfaction. Further, this approach immediately yields a linear program where both conjunction and joint interpretations are combined, and we can optimize any linear combination of expectations. For details, see Remark 1. The technical device to obtain one linear program is to split the standard variables into several, depending on which subsets of constraints they help to achieve. This causes technical complications that have to be dealt with methods of conditional probability.

Related work. The study of Markov decision processes with multiple expectation objectives has been initiated in the area of applied probability theory, where it is known as *constrained MDPs* [Put94,Alt99]. The attention in the study of constrained MDPs has been focused mainly to restricted classes of MDPs, such as unichain MDPs where all states are visited infinitely often under any strategy. Such restriction guarantees the existence of memoryless optimal strategies. The more general problem of MDPs with multiple mean-payoff objectives was first considered in [Cha07] and a complete picture was presented in [BBC$^+$14]. The expectation and satisfaction semantics was considered in [BBC$^+$14], and our work unifies the two different semantics for MDPs. For general MDPs, [CMH06,CFW13] studied MDPs with multiple discounted reward functions. MDPs with multiple qualitative ω-regular specifications were studied in [EKVY08]. It was shown that the Pareto curve can be approximated in polynomial time; the algorithm reduces the problem to MDPs with multiple reachability specifications, which can be solved by multi-objective linear programming [PY00]. In [FKN$^+$11], the results of [EKVY08] were extended to combine ω-regular and expected total reward objectives. The problem of multiple percentile queries (conjunctive satisfaction) has been considered for various objectives, such as mean-payoff, limsup, liminf, shortest path in [RRS14]. However, [RRS14] does not consider optimizing the expectation, whereas we consider maximizing expectation along with satisfaction semantics. The notion of risks has been considered in MDPs with discounted objectives [WL99], where the goal is to maximize (resp., minimize) the probability (risk) that the expected total

discounted reward (resp., cost) is above (resp., below) a threshold. The notion of strong risk aversion, where for risk the probabilistic choices are treated as an adversary was considered in [BFRR14]. In [BFRR14] the problem was considered for single reward for mean-payoff and shortest path. In contrast, we consider risk aversion with probabilistic guarantee, for multiple reward functions. Moreover, since [BFRR14] generalizes mean-payoff games, no polynomial-time solution is known, whereas in our case, we present polynomial-time algorithms for the single reward case and in almost all cases of multiple rewards (see the second item of our contributions).

2 Preliminaries

2.1 Basic definitions

We mostly follow the basic definition of [BBC+14] with only minor deviations. We use $\mathbb{N}, \mathbb{Q}, \mathbb{R}$ to denote the sets of positive integers, rational and real numbers, respectively. For $n \in \mathbb{N}$, we denote $[n] = \{1, \ldots, n\}$. Given two vectors $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^k$, where $k \in \mathbb{N}$, we write $\boldsymbol{v} \geq \boldsymbol{w}$ iff $\boldsymbol{v}_i \geq \boldsymbol{w}_i$ for all $1 \leq i \leq k$. The set of all distributions over a countable set X is denoted by $\mathit{dist}(X)$. Further, $d \in \mathit{dist}(X)$ is Dirac if $d(x) = 1$ for some $x \in X$.

Markov chains. A *Markov chain* is a tuple $M = (L, P, \mu)$ where L is a countable set of locations, $P : L \to \mathit{dist}(L)$ is a probabilistic transition function, and $\mu \in \mathit{dist}(L)$ is the initial probability distribution.

A *run* in M is an infinite sequence $\omega = \ell_1 \ell_2 \cdots$ of locations, a *path* in M is a finite prefix of a run. Each path w in M determines the set $\mathsf{Cone}(w)$ consisting of all runs that start with w. To M we associate the probability space $(\mathsf{Runs}, \mathcal{F}, \mathbb{P})$, where Runs is the set of all runs in M, $\mathcal{F}$ is the σ-field generated by all $\mathsf{Cone}(w)$, and $\mathbb{P}$ is the unique probability measure such that $\mathbb{P}(\mathsf{Cone}(\ell_1, \ldots, \ell_k)) = \mu(\ell_1) \cdot \prod_{i=1}^{k-1} P(\ell_i)(\ell_{i+1})$.

Markov decision processes. A *Markov decision process* (MDP) is a tuple $G = (S, A, \mathit{Act}, \delta, \hat{s})$ where S is a finite set of states, A is a finite set of actions, $\mathit{Act} : S \to 2^A \setminus \{\emptyset\}$ assigns to each state s the set $\mathit{Act}(s)$ of actions enabled at s so that $\{\mathit{Act}(s) \mid s \in S\}$ is a partitioning of A, $\delta : A \to \mathit{dist}(S)$ is a probabilistic transition function that given a state s and an action $a \in \mathit{Act}(s)$ enabled at s gives a probability distribution over the successor states, and $\hat{s}$ is the initial state. Note that we consider that every action is enabled in exactly one state.

A *run* in G is an infinite alternating sequence of states and actions $\omega = s_1 a_1 s_2 a_2 \cdots$ such that for all $i \geq 1$, $a_i \in \mathit{Act}(s_i)$ and $\delta(a_i)(s_{i+1}) > 0$. We denote by Runs_G the set of all runs in G. A *path* of length k in G is a finite prefix $w = s_1 a_1 \cdots a_{k-1} s_k$ of a run in G.

Strategies and plays. Intuitively, a strategy in an MDP G is a "recipe" to choose actions. Usually, a strategy is formally defined as a function $\sigma : (SA)^* S \to \mathit{dist}(A)$ that given a finite path w, representing the history of a play, gives a probability distribution over the actions enabled in the last state. In this paper, we adopt a somewhat different (though equivalent—see [BBC+14, Section 6])

definition, which is more convenient for our setting. Let M be a countable set of *memory elements*. A *strategy* is a triple $\sigma = (\sigma_u, \sigma_n, \alpha)$, where $\sigma_u : A \times S \times \mathsf{M} \to \mathit{dist}(\mathsf{M})$ and $\sigma_n : S \times \mathsf{M} \to \mathit{dist}(A)$ are *memory update* and *next move* functions, respectively, and α is an initial distribution on memory elements. We require that for all $(s, m) \in S \times \mathsf{M}$, the distribution $\sigma_n(s, m)$ assigns a positive value only to actions enabled at s.

A *play* of G determined by a strategy σ is a Markov chain G^σ where the set of locations is $S \times \mathsf{M} \times A$, the initial distribution μ is zero except for $\mu(\hat{s}, m, a) = \alpha(m) \cdot \sigma_n(\hat{s}, m)(a)$, and

$$P(s, m, a)(s', m', a') \;=\; \delta(a)(s') \cdot \sigma_u(a, s', m)(m') \cdot \sigma_n(s', m')(a')$$

Hence, G^σ starts in a location chosen randomly according to α and σ_n. In a current location (s, m, a), the next action to be performed is a, hence the probability of entering s' is $\delta(a)(s')$. The probability of updating the memory to m' is $\sigma_u(a, s', m)(m')$, and the probability of selecting a' as the next action is $\sigma_n(s', m')(a')$. Note that these choices are independent, and thus obtain the product above. The induced probability measure is denoted by $\mathbb{P}^\sigma$ and "almost surely" or "almost all runs" refers to happening with probability 1 according to this measure. The respective expected value of a random variable X is $\mathbb{E}^\sigma[f] = \int_{\mathsf{Runs}} f \; d\,\mathbb{P}^\sigma$. For $t \in \mathbb{N}$, random variables S_t, A_t return s, a, respectively, where (s, m, a) is the t-th location on the run.

Strategy types. In general, a strategy may use infinite memory, and both σ_u and σ_n may randomize. The strategy is

- *deterministic-update*, if α is Dirac and the memory update function gives a Dirac distribution for every argument;
- *deterministic*, if it is deterministic-update and the next move function gives a Dirac distribution for every argument.

A *stochastic-update* strategy is a strategy that is not necessarily deterministic-update and *randomized* strategy is a strategy that is not necessarily deterministic. We also classify the strategies according to the size of memory they use. Important subclasses are *memoryless* strategies, in which M is a singleton, *n-memory* strategies, in which M has exactly n elements, and *finite-memory* strategies, in which M is finite.

End components. A set $T \cup B$ with $\emptyset \neq T \subseteq S$ and $B \subseteq \bigcup_{t \in T} \mathit{Act}(t)$ is an *end component* of G if (1) for all $a \in B$, whenever $\delta(a)(s') > 0$ then $s' \in T$; and (2) for all $s, t \in T$ there is a path $\omega = s_1 a_1 \dots a_{k-1} s_k$ such that $s_1 = s$, $s_k = t$, and all states and actions that appear in ω belong to T and B, respectively. An end component $T \cup B$ is a *maximal end component (MEC)* if it is maximal with respect to subset ordering. Given an MDP, the set of MECs is denoted by MEC.

For a finite-memory strategy σ, a *bottom strongly connected component* (BSCC) of G^σ is a subset of locations $W \subseteq S \times \mathsf{M} \times A$ such that (i) for all $\ell_1 \in W$ and $\ell_2 \in S \times \mathsf{M} \times A$, if there is a path from ℓ_1 to ℓ_2 then $\ell_2 \in W$, and (ii) for all $\ell_1, \ell_2 \in W$ we have a path from ℓ_1 to ℓ_2. Every BSCC W determines a unique end component $\{s \mid (s, m, a) \in W\} \cup \{a \mid (s, m, a) \in W\}$ of G, and we sometimes do not strictly distinguish between W and its associated end component.

2.2 Problem statement

In order to define our problem, we first briefly recall how long-run average can be defined. Let $G = (S, A, Act, \delta, \hat{s})$ be an MDP, $n \in \mathbb{N}$ and $\boldsymbol{r} : A \to \mathbb{Q}^n$ an n-dimensional *reward function*. Since the random variable given by the limit-average function $\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{r}(A_t)$ may be undefined for some runs, we consider maximizing the respective pointwise limit inferior:

$$\mathrm{lr}_{\inf}(\boldsymbol{r}) = \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{r}(A_t)$$

i.e. $\mathrm{lr}_{\inf}(\boldsymbol{r})(\omega)_i = \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{r}(A_t(\omega))_i$ for each $i \in [n]$. Similarly, we could define $\mathrm{lr}_{\sup}(\boldsymbol{r}) = \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{r}(A_t)$. However, maximizing limit superior is less interesting, see [BBC+14].

This paper is concerned with the following tasks:

Realizability (multi-quant-conjunctive): Given an MDP, $n \in \mathbb{N}, \boldsymbol{r} : A \to \mathbb{Q}^n, \boldsymbol{exp} \in \mathbb{R}^n, \boldsymbol{sat} \in \mathbb{R}^n, \boldsymbol{pr} \in [0,1]^n$, decide whether there is a strategy σ such that $\forall i \in [n]$

- $\mathbb{E}^\sigma[\mathrm{lr}_{\inf}(\boldsymbol{r})_i] \geq \boldsymbol{exp}_i$. (EXP)
- $\mathbb{P}^\sigma[\mathrm{lr}_{\inf}(\boldsymbol{r})_i \geq \boldsymbol{sat}_i] \geq \boldsymbol{pr}_i$, (conjunctive-SAT)

Witness strategy synthesis: If realizable, construct a strategy satisfying the requirements.

ε-witness strategy synthesis: If realizable, construct a strategy satisfying the requirements with $\boldsymbol{exp} - \varepsilon \cdot \mathbf{1}$ and $\boldsymbol{sat} - \varepsilon \cdot \mathbf{1}$.

We are also interested in **(multi-quant-joint)** a variant of **(multi-quant-conjunctive)** where (conjunctive-SAT) is replaced by

$$\mathbb{P}^\sigma[\mathrm{lr}_{\inf}(\boldsymbol{r}) \geq \boldsymbol{sat}] \geq pr \qquad \text{(joint-SAT)}$$

for $pr \in [0,1]$. Further, we consider the following special cases:

(multi-qual)	$\boldsymbol{pr} = \mathbf{1}$
(mono-quant)	$n = 1$
(mono-qual)	$n = 1, \boldsymbol{pr} = 1$

The relationship between the problems is depicted in Fig. 1.

2.3 Example

Example 1 (running). We illustrate **(multi-quant-conjunctive)** with an MDP of Fig. 2 with $n = 2$, rewards as depicted and $\boldsymbol{exp} = (1.1, 0.5), \boldsymbol{sat} = (0.5, 0.5), \boldsymbol{pr} = (0.8, 0.8)$. Observe that rewards of actions ℓ and r are irrelevant as these action can almost surely be taken only finitely many times.

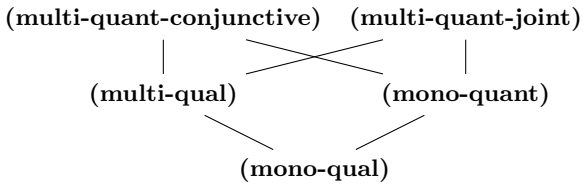

Fig. 1: Relationship of the defined problems with lower problems being specializations of the higher ones

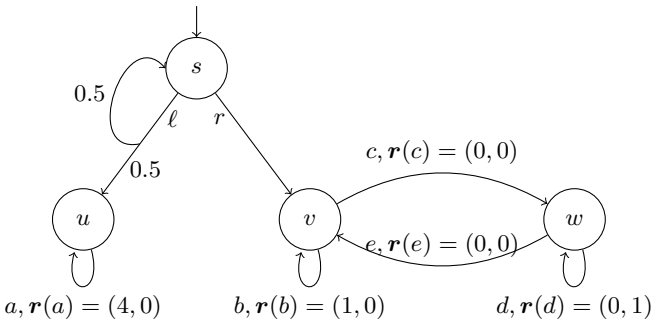

Fig. 2: An MDP with two-dimensional rewards

The problem is realizable and the witness strategy has the following properties. Firstly, due to $\boldsymbol{pr}$, with at least 0.6 runs have to jointly exceed the value thresholds $(0.5, 0.5)$. This is only possible in the right MEC by playing each b and d half of the time and switching between them with a decreasing frequency, so that the frequency of c, e is in the limit 0. Secondly, with 0.2 we reach the left MEC and play a. Thirdly, with 0.2 we reach again the right MEC but only play d with frequency 1. In order to play these three kinds of runs, in the first step in s we take ℓ with probability 0.4 and r with 0.6, and if we return back to s we play r with probability 1. If we reach the MEC on the right, we toss a biased coin and with 0.25 we go to w and play the third kind of runs, and with 0.75 play the first kind of runs.

Observe that although both the expectation and value threshold for the second reward are 0.5, the only solution is not to play all runs with this rewards, but some with a lower one and some with a higher one. Also note that each of the three types of runs must be present in any witness strategy. Most importantly, in the MEC state w we have to play in two different ways, depending on which subset of value thresholds we intend to satisfy.

3 Our solution

In this section, we briefly recall the solution of a previously considered problem and show our solution to the general **(multi-quant-conjunctive)** realizability problem, along with an overview of the correctness proof. For **(multi-quant-joint)** and a detailed analysis of the special cases and the respective complexities, see Section 4.

3.1 Previous results

In [BBC$^+$14], a solution to a special case with only the (EXP) constraint has been given. The existence of a witnessing strategy was shown equivalent to the existence of a solution of the linear program in Fig. 3.

Requiring all variables y_a, y_s, x_a for $a \in A, s \in S$ be non-negative, the program is the following:

1. transient flow: for $s \in S$
$$\mathbf{1}_{s_0}(s) + \sum_{a \in A} y_a \cdot \delta(a,s) = \sum_{a \in Act(s)} y_a + y_s$$
2. almost-sure switching to recurrent behaviour:
$$\sum_{s \in C \in \mathsf{MEC}} y_s = 1$$
3. probability of switching in a MEC is the frequency of using its actions: for $C \in \mathsf{MEC}$
$$\sum_{s \in C} y_s = \sum_{a \in C} x_a$$
4. recurrent flow: for $s \in S$
$$\sum_{a \in A} x_a \cdot \delta(a,s) = \sum_{a \in Act(s)} x_a$$
5. expected rewards:
$$\sum_{a \in A} x_a \cdot \boldsymbol{r} \geq \boldsymbol{exp}$$

Fig. 3: Linear program of [BBC$^+$14] for (EXP)

Intuitively, x_a is the expected frequency of using a on the long run; Equation 4 thus expresses the flow in MECs and Equation 5 the expected long run reward. Variables y_a are the expected number of using a until we switch to the recurrent behaviour in MECs and y_s is the probability of this switch upon reaching s; Equation 1 thus expresses the flow before switching. To relate these variables,

Equation 3 states that the probability to switch within a given MEC is the same whether viewed from the transient or recurrent flow perspective. Actually, one could eliminate variables y_s and use directly x_a in Equation 1 and leave out Equation 3 completely, in the spirit of [Put94]. However, the form with explicit y_s is sometimes more convenient. Finally, Equation 2 states that switching happens almost surely. Note that summing Equation 1 over all $s \in S$ yields $\sum_{s \in S} y_s = 1$. Since y_s can be shown to equal 0 for state s not in MEC, Equation 2 is redundant.

Further, apart from considering (EXP) separately, [BBC+14] also considers the constraint (joint-SAT) separately. While the former was solved using the linear program above, the latter required a reduction to one linear program per each MEC and another one to combine the results. We shall provide one linear program for the general problems, thus unifying the results.

3.2 Our general solution

There are two main tricks to incorporate the satisfaction semantics. The first one is to ensure that a flow exceeds the value threshold:

Solution to (multi-qual) When the additional constraint (SAT) is added so that almost all runs satisfy $\mathrm{lr}_{\inf}(\boldsymbol{r}) \geq \boldsymbol{sat}$, then the linear program of Fig. 3 shall be extended with the following additional equation:

6. almost-sure satisfaction: for $C \in \mathsf{MEC}$

$$\sum_{a \in C} x_a \cdot \boldsymbol{r}(a) \geq \sum_{a \in C} x_a \cdot \boldsymbol{sat}$$

Note that x_a represents the absolute frequency of playing a (not relative within the MEC). Intuitively, Equation 6 thus requires in each MEC the average reward be at least $\boldsymbol{sat}$. Here we rely on the non-trivial fact proved later, that in a MEC, actions can be played on almost all runs with the given frequencies for any flow.

The second trick ensures that each conjunct in the satisfaction constraint can be handled separately and the probability threshold checked.

Solution to (multi-quant) When each value threshold $\boldsymbol{sat}_i$ comes with a non-trivial probability threshold $\boldsymbol{pr}_i$, some runs may and some may not have the long-run reward exceeding $\boldsymbol{sat}_i$. In order to speak about each group, we split the set of runs for each reward into parts which do and which do not exceed the threshold.

Technically, we keep Equations 1–5 as well as 6, but split x_a into $x_{a,N}$ for $N \subseteq [n]$, where N describes the subset of exceeded thresholds; similarly for y_s. The linear program L then takes the form displayed in Fig. 4.

Intuitively, only the runs in the appropriate "N-class" are required in Equation 6 to have rewards exceeding the threshold. However, only the appropriate "N-classes" are considered for surpassing the probabilistic threshold in Equation 7.

Requiring all variables $y_a, y_{s,N}, x_{a,N}$ for $a \in A, s \in S, N \subseteq [n]$ be non-negative, the program is the following:

1. transient flow: for $s \in S$
$$\mathbf{1}_{s_0}(s) + \sum_{a \in A} y_a \cdot \delta(a,s) = \sum_{a \in Act(s)} y_a + \sum_{N \subseteq [n]} y_{s,N}$$
2. almost-sure switching to recurrent behaviour:
$$\sum_{\substack{s \in C \in \mathsf{MEC} \\ N \subseteq [n]}} y_{s,N} = 1$$
3. probability of switching in a MEC is the frequency of using its actions: for $C \in \mathsf{MEC}, N \subseteq [n]$
$$\sum_{s \in C} y_{s,N} = \sum_{a \in C} x_{a,N}$$
4. recurrent flow: for $s \in S, N \subseteq [n]$
$$\sum_{a \in A} x_{a,N} \cdot \delta(a,s) = \sum_{a \in Act(s)} x_{a,N}$$
5. expected rewards:
$$\sum_{\substack{a \in A, \\ N \subseteq [n]}} x_{a,N} \cdot \boldsymbol{r}(a) \geq \boldsymbol{exp}$$
6. commitment to satisfaction: for $C \in \mathsf{MEC}$, $N \subseteq [n]$, $i \in N$
$$\sum_{a \in C} x_{a,N} \cdot \boldsymbol{r}(a)_i \geq \sum_{a \in C} x_{a,N} \cdot \boldsymbol{sat}_i$$
7. satisfaction: for $i \in [n]$
$$\sum_{\substack{a \in A, \\ N \subseteq [n] : i \in N}} x_{a,N} \geq \boldsymbol{pr}_i$$

Fig. 4: Linear program L for **(multi-quant-conjunctive)**

Theorem 1. *Given a realizability problem, the respective system L satisfies the following:*

1. *The system L is constructible and solvable in time polynomial in the size of G and exponential in n.*
2. *Every witness strategy induces a solution to L.*
3. *Every solution to L effectively induces a witness strategy.*

Example 2 (running). The linear program L for Example 1 is depicted in full in Appendix **??**. Here we spell out only several points: Equation 1 for state s

$$1 + 0.5y_\ell = y_\ell + y_r + y_{s,\emptyset} + y_{s,\{1\}} + y_{s,\{2\}} + y_{s,\{1,2\}} \tag{1}$$

expresses the Kirchhoff's law for the flow through the initial state. Equation 6 for the MEC $C = \{v, w, a, b, c, d\}$, $N = \{1, 2\}$, $i = 1$

$$x_{b,\{1,2\}} \cdot 1 \geq (x_{b,\{1,2\}} + x_{c,\{1,2\}} + x_{d,\{1,2\}} + x_{e,\{1,2\}}) \cdot 0.5 \tag{2}$$

expresses that runs ending up in C and satisfying both satisfiability thresholds have to use action b at least half of the time. The same holds for d and thus actions c, e must be played with zero frequency on these runs. Equation 7 for $i = 1$ sums up the use of all action on runs that have committed to exceed the probability threshold either for the first reward, or for the first *and* the second reward.

3.3 Proof overview

The first point follows immediately from the syntax of L and the existence of a polynomial algorithm for linear programming [Kar84].

The second point is proven in Appendix A. The proof method roughly follows that of [BBC+14, Proposition 4.5]. Given a winning strategy σ, we construct values for variables so that a valid solution is obtained. The technical details can be found in Section 3.5 and Appendix A.

The proof of [BBC+14] sets the values of x_a to be the expected frequency of using a by σ, i.e.

$$\mathrm{freq}^\sigma(a) := \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{P}^\sigma[A_t = a]$$

Since this Cesaro limit may not be defined, a suitable value $f(a)$ between the limit inferior and superior has to be taken. In contrast to the approach of [BBC+14], we need to distinguish among runs exceeding various subsets of the value thresholds $\boldsymbol{sat}_i, i \in [n]$. For $N \subseteq [n]$, we call a run *N-good* if $\mathrm{lr}_{\inf}(\boldsymbol{r})_i \geq \boldsymbol{sat}_i$ for exactly $i \in N$. Now instead of examining frequencies $f(a)$ of each action a, we examine frequencies $f_N(a)$ of action a on *N-good* runs separately, for each N.

Example 3 (running). The strategy of Example 1 induces the following x-values. For instance, action a is played with a frequency 1 on runs of measure 0.2, hence $x_{a,\{1\}} = 0.2$ and $x_{a,\emptyset} = x_{a,\{2\}} = x_{a,\{1,2\}} = 0$. Action d is played with frequency 0.5 on runs of measure 0.6 exceeding both value thresholds, and with frequency 1 on runs of measure 0.2 exceeding only the second value thresholds. Consequently, $x_{d,\{1,2\}} = 0.3$ and $x_{d,\{2\}} = 0.2$ whereas $x_{d,\emptyset} = x_{d,\{1\}} = 0$.

Values for y-variables are derived from the expected number of using actions during the "transient" behavior of the strategy. Since the expectation may be infinite in general, an equivalent strategy is constructed, which is memoryless in the transient part, but switches to the recurrent behaviour in the same way. Then the expectations are finite and the result of [EKVY08] yields values satisfying the transient flow equation. Further, similarly as for x-values, instead of simply switching to recurrent behaviour in a particular MEC, we consider switching in a MEC *and* the set N for which the following recurrent behaviour is N-good.

Example 4 (running). The strategy of Example 1 plays in s for the first time ℓ with probability 0.4 and r with 0.6, and next time r with probability 1. This is equivalent to a memoryless strategy playing ℓ with 1/3 and r with 2/3. Indeed, both ensure reaching the left MEC with 0.2 and the right one with 0.8. Therefore, we the expected number of playing r is

$$y_r = \frac{2}{3} + \frac{1}{6} \cdot \frac{2}{3} + \left(\frac{1}{6}\right)^2 \cdot \frac{2}{3} + \cdots = \frac{5}{6}$$

The values $y_{u,\{1\}} = 0.2$, $y_{v,\{1,2\}} = 0.6$, $y_{v,\{2\}} = 0.2$ are given by the probability measures of each "kind" (see Ex. 1) of runs.

The third point is proven in Appendix B. Given a solution to L, we construct a winning strategy σ, which has a particular structure. The technical details can be found in Section 3.6 and Appendix B. The general pattern follows the proof method of [BBC$^+$14, Proposition 4.5], but there are several important differences.

First, a strategy is designed to behave in a MEC so that the frequencies of actions match the x-values. The structure of the proof differs here and we focus on underpinning the following key principle. Even if the flow described by x-variables has several disconnected components within the MEC, and thus actions connecting them must not be played with positive frequency, there are still strategies that on almost all runs play actions of all components with exactly the given frequencies. The trick is to play the "connecting" actions with an increasingly negligible frequency. As a result, the strategy induces only one BSCC, which simplifies matters and allows us to prove that stochastic update is not necessary. Therefore, the deterministic update is sufficient, in particular also for (joint-SAT), which improves the strategy complexity known from [BBC$^+$14].

Second, the construction of the recurrent part of the strategy as well as switching to it has to reflect again the different parts of L for different N, resulting in N-good behaviours.

Example 5 (running). A solution with $x_{b,\{1,2\}} = 0.3, x_{d,\{1,2\}} = 0.3$ induces two disconnected flows. Each is an isolated loop, yet we can play a strategy that plays both actions exactly half of the time. We discuss this in Section 3.6 where we investigate the construction of the strategy from the solution in more details, necessary for later complexity discussion.

3.4 Important aspects of our approach and its consequences

Remark 1. We now explain some important conceptual aspects of our result. The previous proof idea from [BBC+14] is as follows: (1) The problem for expectation semantics is solved by a linear program. (2) The problem for satisfaction semantics is solved as follows: each MEC is considered, solved separately using a linear program, and then a reachability problem is solved using a different linear program. In comparison, our proof has two conceptual steps. Since our goal is to optimize the expectation (which intuitively requires a linear program), the first step is to come up with a single linear program for the satisfaction semantics. The second step is to come up with a linear program that unifies the linear program for expectation semantics and the linear program for satisfaction semantics, and presents a solution to our problem.

Since our solution captures all the frequencies and percentiles within one linear program, we can work with all the flows at once. This has several consequences:

- While all the hard constraints are given as a part of the problem, we can easily find maximal solution with respect to e.g. weighted reward expectation, i.e. $\boldsymbol{w} \cdot \mathrm{lr}_{\inf}(\boldsymbol{r})$, as it can be expressed as $\boldsymbol{w} \cdot \sum_{a,N} x_{a,N} \cdot \boldsymbol{r}(a)$, where $\boldsymbol{w}$ is the vector of weights for each reward dimension. This is also relevant for the construction of the Pareto curve.
- We can easily express constraints for multiple and joint constraints $\mathbb{P}^\sigma\left[\bigwedge_{k_i} \mathrm{lr}_{\inf}(\boldsymbol{r}_{k_i}) \geq pr\right]$ by adding a copy of Equation 7 for arbitrary subsets N of constraints.
- The number of variables used in the linear program immediately yields an upper bound on the computational complexity of various subclasses of the general problem. Several polynomial bounds are proven in Section 4.

3.5 Technical proof of Theorem 1, item 2

We prove that every witness strategy ϱ induces a solution to L. We start with constructing values for variables x_a. In general, the frequencies $\mathrm{freq}^\varrho(a)$ of the actions may not be well defined, because the defining limits may not exist. Further, it may be unavoidable to have different frequencies for several sets of runs of positive measure. There are two tricks to overcome this difficulty. Firstly, we partition the runs into several classes depending on which parts of the objective they achieve. Secondly, within each class we pick suitable values lying between $\mathrm{lr}_{\inf}(\boldsymbol{r})$ and $\mathrm{lr}_{\sup}(\boldsymbol{r})$ of these runs.

Formally, for $N \subseteq [n]$, let

$$\Omega_N = \{\omega \in \mathit{Runs} \mid \forall i \in N : \mathrm{lr}_{\inf}(\boldsymbol{r})_i(\omega) \geq \boldsymbol{sat}(i) \wedge \forall i \notin N : \mathrm{lr}_{\inf}(\boldsymbol{r})_i(\omega) < \boldsymbol{sat}(i)\}$$

Then Ω_N for $N \subseteq [n]$ form a paritioning of *Runs*. We define $f_N(a)$, lying between $\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{P}^{\varrho}[A_t = a \mid \Omega_N]$ and $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{P}^{\varrho}[A_t = a \mid \Omega_N]$, which can be safely substituted for $x_{a,N}$ in L. Since every infinite sequence contains an infinite convergent subsequence, there is an increasing sequence of indices, $T_0, T_1, \ldots$, such that the following limit exists for each action $a \in A$

$$f_N(a) \coloneqq \lim_{\ell\to\infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^{\varrho}[A_t = a \mid \Omega_N] \cdot \mathbb{P}^{\varrho}[\Omega_N]$$

We set $x_{a,N} \coloneqq f_N(a)$ for all $a \in A$ and $N \subseteq [n]$ (where $x_{a,N} = 0$ whenever $\mathbb{P}^{\varrho}[\Omega_N] = 0$). In Appendix A, we prove Equation 4–7 are satisfied.

Now we set the values for y_χ, $\chi \in A \cup S \times 2^{[n]}$, One could obtain the values y_χ using the methods of [Put94, Theorem 9.3.8], which requires the machinery of deviation matrices. Instead, we can first simplify the behaviour of ϱ in the transient part to memoryless using [BBC+14] and then obtain y_χ directly, like in [EKVY08], as expected numbers of taking actions. To this end, for a state s we define $\Diamond s$ to be the set of runs that contain s.

Similarly to [BBC+14, Proposition 4.2 and 4.5], we modify the MDP G into another MDP $\bar{G}$ as follows: For each $s \in S, N \subseteq [n]$, we add a new absorbing state $f_{s,N}$. The only available action for $f_{s,N}$ leads to a loop transition back to $f_{s,N}$ with probability 1. We also add a new action, $a_{s,N}$, to every $s \in S$ for each $N \subseteq [n]$. The distribution associated with $a_{s,N}$ assigns probability 1 to $f_{s,N}$. Finally, we remove all unreachable states. The construction of [BBC+14] is the same but only a single value is used for N.

Claim 1. There is a strategy $\bar{\varrho}$ in $\bar{G}$ such that for every $C \in \mathsf{MEC}$ and $N \subseteq [n]$,

$$\sum_{s\in C} \mathbb{P}^{\bar{\varrho}}[\Diamond f_{s,N}] = \mathbb{P}^{\varrho}[\Omega_C \cap \Omega_N]$$

By [EKVY08, Theorem 3.2], there are values y_χ satisfying the following:

- Equation 1 is satisfied. Further, summing up Equation 1 for each s yields Equation 2.
- $y_{s,N} \geq \sum_{s\in C} \mathbb{P}^{\bar{\varrho}}[\Diamond f_{s,N}]$. By Claim 6 for each $C \in \mathsf{MEC}$ we thus have

  $$\sum_{s\in C} y_{s,N} \geq \mathbb{P}^{\varrho}[\Omega_C \cap \Omega_N]$$

  and summing up over all C and N we have

  $$\sum_{N\subseteq[n]} \sum_{s\in S} y_{s,N} \geq \sum_{N\subseteq[n]} \mathbb{P}^{\varrho}[\Omega_N]$$

  where the first term is 1 by Equation 2, the second term is 1 by partitioning of Runs, hence they are actually equal and thus

  $$\sum_{s\in C} y_{s,N} = \mathbb{P}^{\varrho}[\Omega_C \cap \Omega_N] = \sum_{a\in C} x_{a,N}$$

  where the last equality follows by Claim 5, yielding Equation 3.

– There is a memoryless strategy $\hat{\varrho}$ such that $\mathbb{P}^{\hat{\varrho}}[\Diamond f_{s,N}] = \mathbb{P}^{\bar{\varrho}}[\Diamond f_{s,N}]$. The value y_a is the expected number of taking a by $\hat{\varrho}$ (for actions a preserved in $\bar{G}$) and $y_{s,N} = \mathbb{P}^{\hat{\varrho}}[\Diamond f_{s,N}]$. By [EKVY08, Lemma 3.3] all y_a and $y_{s,N}$ are indeed finite values.

and prove that they satisfy Equations 1–3 of L when the values $f_N(a)$ are assigned to $x_{a,N}$.

3.6 Technical proof of Theorem 1, item 3: Description of a witness strategy induced by a solution to L

Every solution to L effectively induces a witness strategy.. Here we investigate the construction of the strategy from the solution in more details. This will be necessary for establishing the complexity bounds in Section 4.

We start with the recurrent part. We prove that even if the flow of Equation 4 is "disconnected" we may still play the actions with the exact frequencies $x_{a,N}$ on almost all runs (here $N \subseteq [n]$ is fixed for now).

Firstly, we construct a strategy for each "strongly connected" part of the solution $x_{a,N}$ and connect the parts, thus averaging the frequencies. This happens at a cost of small error used for transiting between the strongly connected parts.

Claim 2. In a strongly connected MDP, for every $\varepsilon > 0$ there is a strategy ζ_ε such that almost all runs ω satisfy that $\mathrm{freq}^{\zeta_\varepsilon}(a)(\omega)$ is positive for all $a \in A$ and

$$\mathrm{freq}^{\zeta_\varepsilon}(a)(\omega) > x_{a,N} / \sum_a x_{a,N} - \varepsilon$$

Secondly, we eliminate this error as we let the transiting happen with mass vanishing over time.

Claim 3. In a strongly connected MDP, let ξ_i be a sequence of strategies with each freq^{ξ_i} constant for almost all conforming runs and positive, such that $\lim_{i\to\infty} \mathrm{freq}^{\xi_i}$ is well defined. Then there is a strategy ξ with

$$\mathrm{freq}^{\xi} = \lim_{i\to\infty} \mathrm{freq}^{\xi_i}$$

which is constant on almost all runs.

In any MEC, we can now define the strategy ξ_N such that almost all runs ω satisfy for each $a \in A$

$$\mathrm{freq}^{\xi_N}(a)(\omega) = x_a / \sum_a x_a$$

using Claim 3 taking ξ_i to be $\zeta_{1/i}$ from Claim 2. Note that as a consequence, all actions and states in a MEC are visited infinitely often. This will be later useful for the strategy complexity analysis.

The reward of ξ_N is for almost all runs

$$\mathrm{lr}_{\inf}(\boldsymbol{r})(\omega) = \sum_a \big(\bar{x}_{a,N} \cdot \boldsymbol{r}(a)\big) / \sum_a \bar{x}_{a,N}$$

This property of the strategy holds for almost all runs, which is a stronger statement than in [BBC+14] and we need it to combine the satisfaction requirements. Finally, note that the strategy uses infinite memory, but only needs a counter and to know the current state.

We now consider the transient part of the solution that plays ξ_N's with various probabilities. Let "switch to ξ_N in C" denote the event that a strategy updates its memory, while in C, into such element that it starts playing exactly as ξ_N. We can stich all ξ_N's together as follows:

Claim 4. Given strategies $\{\xi_N\}$, a non-negative solution $y_a, y_{s,N}$ to Equation 1 and 3 induces a strategy σ such that for every MEC C

$$\mathbb{P}^\sigma[\text{switch to } \xi_N \text{ in } C] = \sum_{a \in C \cap A} \bar{x}_{a,N}$$

In Appendix B we prove this is indeed a witness strategy.

4 Computational and strategy complexity

In this section, we discuss the solutions to all other introduced problems as well as their algorithmic and strategy complexities.

4.1 Computational complexity

(multi-quant-conjunctive) As we have seen, there are $\mathcal{O}(|G| \cdot n) \cdot 2^n$ variables in the linear program L. By Theorem 1, the upper bound on the computational time complexity is polynomial in the number of variables in system L. Hence, the realizability problem can be decided in time polynomial in $|G|$ and exponential in n.

(multi-quant-joint) In order to decide this problem, the only subset of runs to exceed the probability threshold is the set of runs with all long-run rewards exceeding their thresholds, i.e. $\Omega_{[n]}$ (introduced in Section A). The remaining runs need not be partitioned and can be all considered to belong to $\Omega_\emptyset$ without violating any constraint. Intuitively, each $x_{a,\emptyset}$ now stands for the original sum $\sum_{N \subseteq [n]: N \neq [n]} x_{a,N}$; similarly for y-variables. Consequently, the only relevant variables of L are those indexed by N taking values $[n]$ or $\emptyset$. The remaining variables can be left out of the system. Since there are now $\mathcal{O}(|G| \cdot n)$ variables, the problem as well as its special cases can be decided in polynomial time.

(multi-qual) is a qualitative specialization of **(multi-quant)** with $\boldsymbol{pr} = \mathbf{1}$ as well as of **(joint)** with $pr = 1$. Since $\mathbb{P}^\sigma\left[\mathsf{Runs} \setminus \Omega_{[n]}\right] = 0$ for any winning σ, the only relevant index N is $[n]$.

(mono-quant) is a scalar specialization of **(multi-quant)** with $n = 1$ as well as of **(joint)** with $n = 1$. Hence the only relevant indices N are $[n] = \{1\}$ and $\emptyset$.

(mono-qual) is both a scalar specialization of **(multi-qual)** with $n = 1$ and a qualitative specialization of **(mono-quant)** with $pr = 1$. Hence the only relevant index N is $[n] = \{1\}$, thus the index can be removed completely.

Theorem 2. *The **(multi-quant-joint)** realizability problem (and thus also all its special cases) can be decided in time polynomial in $|G|$ and n.*

4.2 Strategy complexity

First, we recall the structure of winning strategies generated from L. In the first phase, a memoryless strategy is applied to reach MECs and switch to the recurrent strategies ξ_N. When the switch takes place, the following two pieces of information are remembered: (1) the binary decision to stay in the current MEC C forever, and (2) the set $N \subseteq [n]$, such that all the produced runs belong to Ω_N. Each recurrent strategy ξ_N is then an infinite-memory strategy, where the memory is simply a counter. The counter determines which memoryless strategy is played.

Due to [BBC+14, Section 5], infinite memory is indeed necessary for (joint-SAT) with $pr = 1$, hence also for **(multi-qual)**.

Further, in our setting where expectation and satisfaction are combined, randomization and memory is necessary even for single reward function (and even for ε-winning strategies).

Example 6. Randomization and memory is necessary for **(mono-quant)** with $\boldsymbol{sat} = 1, \boldsymbol{exp} = 3, \boldsymbol{pr} = 0.55$ and the MDP and $\boldsymbol{r}$ depicted in Fig. 5. We have to remain in MEC $\{s, a\}$ with probability $p \in [0.1, 2/3]$, hence we need a randomized decision. Further, memoryless strategy would either never leave $\{s, a\}$ or would leave it eventually almost surely. The argument holds for ε-winning strategies as well since $1 < 2 < 3 < 0.5 \cdot 0 + 0.5 \cdot 10$.

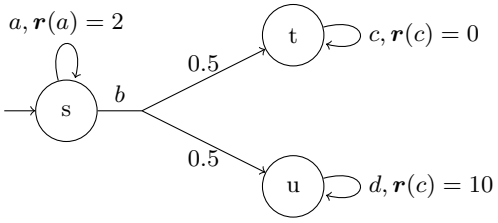

Fig. 5: An MDP with a one-dimensional reward, where randomization and memory is necessary

Example 7. In general, even ε-optimal strategies may require randomization and memory with at least n memory elements. Consider an MDP with a single state s and self-loop a_i with reward $\boldsymbol{r}_i(a_j)$ equal to 1 for $i = j$ and 0 otherwise, for each $i \in [n]$. Fig. 6 illustrates the case of $n = 3$. Further, let $\boldsymbol{sat} = \mathbf{1}$ and $\boldsymbol{pr} = \mathbf{1}/n$. The only way to ε-satisfy the constraints is that for each i, $1/n$ runs take only a_i, but for a negligible portion of time. Since these constraints are mutually incompatible for a single run, n different decisions have to be taken at s, showing the memory requirements. Moreover, as different decisions have to be made at some point with a single history, randomization is necessary.

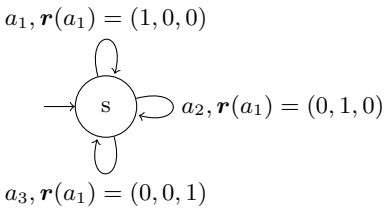

Fig. 6: An MDP where n-memory is necessary, depicted for $n = 3$

As one of the main results, we prove that stochastic update at the moment of switching is not necessary, which improves also the result of [BBC+14].

Lemma 1. *Deterministic update is sufficient for witnessing strategies. Moreover, finite memory is sufficient before switching to ξ_N's.*

Proof (Proof idea). The stochastic decision during the switching in MEC C can be done as a deterministic update after a "toss", a random choice between two actions in C in one of the states of C. Such a toss does not affect the long-run reward as it is only performed finitely many times.

More interestingly, in MECs where no toss is possible, we can remember which states were visited how many times and choose respective probability of leaving or staying in C.

Proof. Let σ be a strategy induced by L as decribed above. We modify it into a strategy ϱ with the same distribution of the long-run rewards. The only stochastic update that σ performs is in a MEC, switching to ξ_N with probability p_N for each N. We modify σ into ϱ in each MEC C separately.

Tossing-MEC case We first assume that there are $\mathit{toss}, a, b \in C$ with $a, b \in \mathit{Act}(\mathit{toss})$. Whenever σ should perform a step in s and possibly make a stochastic-update, say to m_1 with probability p_1 and m_2 with probability p_2, ϱ performs a "toss" instead. A (p_1, p_2)-toss is reaching *toss* with probability 1 (using a memoryless strategy) and taking a, b with probabilities p_1, p_2, respectively. A

deterministic update is made based on the result, in order to remember the result of the toss. After the toss, ϱ returns back to s with probability 1 (again using a memoryless strategy). Now that it already remembers the (p_1, p_2)-toss, it changes the memory accordingly, by a deterministic update.

In general, since the stochastic-update probabilities depend on the action chosen and the state to be entered, we have to perform the toss for each combination before returning to s. Further, whenever there are more possible results for the memory update (various N), then we can use binary encoding of the choices, say with k bits, and repeat the toss with the appropriate probabilities k-times before returning to s.

This can be clearly implemented using a finite memory. Indeed, since there are finitely many states in a MEC and the strategy σ is memoryless, there are only finitely many combinations of tosses to make and remeber till the next simulated update of σ.

Tossfree-MEC case It remains to handle the case when for each state $s \in C$ there is only one action $a \in Act(s) \cap C$. Then all strategies staying in C behave the same here, call this memoryless deterministic strategy γ. Therefore, the only stochastic update that matters is to stay in C or not. The MEC C is left via each action a with the probability

$$leave_a := \sum_{t=1}^{\infty} \mathbb{P}^{\sigma}[A_t = a \mid S_t \in C, S_{t+1} \notin C]$$

and let $\{a \mid leave_a > 0\} = \{a_1, \ldots, a_\ell\}$ be the leaving actions. The strategy ϱ upon entering C performs the following. Subsequently for each $i \in [\ell]$, it first leaves C via a_1 with probability $leave_{a_1}$ (see below how), then via a_2 with probability $\frac{leave_{a_2}}{1-leave_{a_1}}$, and so on via a_i with probability

$$\frac{leave_{a_i}}{1 - \sum_{j=1}^{i-1} leave_{a_j}}$$

After the last attempt with a_ℓ, if we are still in C, we update the memory to stay in C forever (playing γ).

Leaving C via a with probability *leave* can be done as follows. Let $rate = \sum_{s \notin C} \delta(a)(s)$ be the probability to actually leave C when taking a once. Then to achieve the overall probability *leave* of leaving we can reach s with $a \in Act(s)$ and play a with probability 1 and repeat this $\lfloor leave/rate \rfloor$-times and finally reach s once more and play a with probability $leave/rate - \lfloor leave/rate \rfloor$ and the action staying in C with the remaining probability.

In order to implement the strategy in MECs of this second type, for each action it is sufficient to have a counter up to $\lceil 1/p \rceil$, where p is the minimal probability in the MDP. □

Observe that the latter case of the proof is important for satisfaction objectives. It has not been considered in [BBC+14], where deterministic update is shown sufficient only for the expectation objective.

As a consequence of ϱ of Lemma 1 needing only finite memory, we obtain several corollaries. Firstly, infinite memory is only required for winning strategies:

Lemma 2. *Deterministic-update with finite memory is sufficient for ε-witnessing strategies.*

Proof. After switching, the memoryless strategy ζ_ϵ can be played instead of the strategy playing the sequence of ζ_ϵ for decreasing ϵ according to the infinite counter. □

Secondly, infinite memory is only required for multiple rewards.

Lemma 3. *Deterministic-update strategies with finite memory are sufficient for **(mono-quant)**.*

Proof. After switching in C, we can play the following memoryless strategy. In C, there can be several components of the flow. We pick any with the largest long-run reward. □

Further, the construction in the toss-less case gives us a hint for lower bound on memory. We show that for deterministic update witness strategies memory with size dependent on the MDP is needed even for **(mono-quant)**.

Example 8. Memory dependent on the size of transition probabilities is necessary for deterministic-update ε-witnessing strategies for **(mono-quant)**. To this end, let us consider the same realizability problem as in Example 6, but with a slightly modified MDP, depicted in Fig. 7. Again, we have to remain in MEC $\{s, a\}$ with probability $p \in [0.1, 2/3]$ (for ε-witnessing strategies close to these values). Let $\ell > 0$ denote the minimal probability with which any (ε-)witnessing strategy has to leave the MEC and all (ε-)witnessing strategies have to stay in the MEC with positive probability. We show that at least $\lceil \frac{\ell}{\delta} \rceil$-memory is necessary. Observe that this setting also applies to the (EXP) setting of [BBC$^+$14], e.g. $\boldsymbol{exp} = (0.5, 0.5)$ and the MDP in Fig. 8. Therefore, we provide a lower bound also for this simpler case (no lower bound is provided in [BBC$^+$14]).

For a contradiction, assume there are less than $\lceil \frac{\ell}{\delta} \rceil$ memory elements. Then by the pigeon-hole principle, in the first $\lceil \frac{\ell}{\delta} - 1 \rceil$ visits of s, some memory element m appears twice. Note that due to the deterministic updating, each run generates the same play, in particular the same sequence of memory elements. Let p be the probability to eventually leave s provided we are in s with memory m.

If $m = 0$ then the probability to leave s at the start is less $\lceil \frac{\ell}{\delta} - 2 \rceil \cdot \delta < \ell$, a contradiction. Indeed, we have at most $\lceil \frac{\ell}{\delta} - 2 \rceil$ tries to leave s before obtaining memory m and with every try we leave s with probability δ; we conclude by the union bound.

Let $m > 0$. Due to the deterministic updates, all runs staying in s use memory m infinitely often. Since $m > 0$, there is a finite number n of steps such that (1) during these steps the overall probability to leave s is at least $m/2$ and (2) we are using m again. Consequently, the probability of runs staying in s is 0, a contradiction.

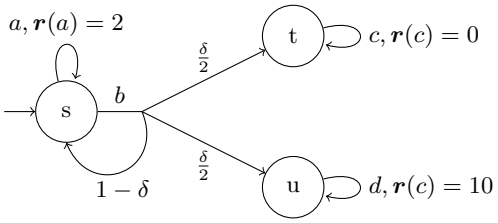

Fig. 7: An MDP with a one-dimensional reward, where memory with size dependent on transition probabilities is necessary for deterministic-update strategies

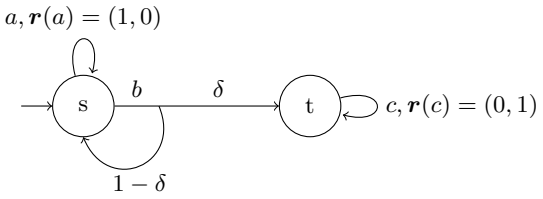

Fig. 8: An MDP family where n-memory is not sufficient for deterministic-update strategies even for (EXP) studied in [BBC+14]

We can even bound the memory with a fixed number in the single-reward case. However, for quantitative satisfaction, we require stochastic update.

Lemma 4. *Stochastic-update 2-memory strategies are sufficient for* ***(mono-quant)****.*

Proof. As in the previous lemma, we can achieve optimal value in each MEC using a memoryless strategy. The strategy σ of Section 3.6, which reaches the MECs and stays in them with given probability, is memoryless up to the point of switch. □

Lemma 5. *Deterministic memoryless strategies are sufficient for* ***(mono-qual)****.*

Proof. For each MEC, there is a value, which is the maximal long-run reward. This is achievable for all runs in the MEC and using a memoryless strategy ξ. We prune the MDP to remove MECs with values below the threshold ***sat***. The optimal strategy maximizes the expected long-run (single-dimensional) reward. Such a strategy can be picked memoryless [Put94]. Intuitively, in this case each MEC is either stayed at almost surely or left almost surely if the value of the outgoing action is higher. □

We summarize the upper and lower bounds on the strategy complexity.

Theorem 3. *The bounds on the complexity of the witnessing strategies are as shown in Table 1.*

Table 1: Complexity results for each of the discussed cases. Upper and lower bounds can be found in the upper and the lower lines, respectively. Results without reference are induced by the specialization/generalization relation depicted in Fig. 1.
Here "inf." means infinite memory consisting of a counter together with a finite memory [Section 3.6]

Case	Computational c.	Witness strategy complexity	ε-witness strategy complexity
(multi-quant-conj.)	$poly(\|G\|, 2^n)$ [Thm.1]	det.-up. [Lem.1] inf.	det.-up. fin.[Lem.2]
	?	rand. inf.	rand. n-mem., for det.-up. G-mem.
(multi-quant-joint)	$poly(\|G\|, n)$ [Thm.2]	det.-up. inf.	det.-up. fin.
		rand. inf.	rand. n-mem., for det.-up. G-mem.
(multi-qual)	$poly(\|G\|, n)$	det.-up. inf.	det.-up. fin.
		rand. inf. [BBC$^+$14, Sec.5]	rand. n-mem. [Ex.7]
(mono-quant)	$poly(\|G\|, n)$	stoch.-up. 2-mem. [Lem.4], det.-up. fin. [Lem.3]	stoch.-up. 2-mem., det.-up. fin.
		rand. mem., for det.-up. G-mem.	rand. [Ex.6] mem. [Ex.6], for det.-up. G-mem. [Ex.8]
(mono-qual)	$poly(\|G\|, n)$	det. memoryless [Lem.5]	det. memoryless

.

Remark 2. While [BBC$^+$14] shows that deterministic update is sufficient for (EXP), it makes no such a statement for (SAT) (both are special cases of **(multi-quant-joint)**). This was possible since the tossfree-MEC case of Lemma 1 is not relevant for (EXP), where the MEC is either almost surely left or almost surely stayed at. Our result thus improves the upper bound for (SAT).

Further, while [BBC$^+$14] shows that pure memoryless strategies are not sufficient for ε-witnessing (SAT) and without a proof for (EXP), our Example 8 shows both randomization and memory of size dependent on the MDP is necessary for (EXP).

4.3 Pareto curve approximation

Theorem 4. *For $\varepsilon > 0$, an ε-approximation of the Pareto curve for **(multi-quant-conjunctive)** can be constructed in time polynomial in $|G|$ and $\frac{1}{\varepsilon}$ and exponential in n, moreover, for **(multi-quant-joint)** in time polynomial in $|G|$, $\frac{1}{\varepsilon}$, and n.*

Proof. We replace $\boldsymbol{exp}$ in Equation 5 of L by a vector $\boldsymbol{v}$ of variables. Maximizing with respect to $\boldsymbol{v}$ is a multi-objective linear program. By [PY00], we can ε-approximate the Pareto curve in time polynomial in the size of the program, $\frac{1}{\varepsilon}$, and the dimension of $\boldsymbol{v}$.

5 Conclusion

We have provided a unifying solution framework to the expectation and satisfaction optimization of Markov decision processes with multiple rewards. This

allows us to synthesize optimal and ε-optimal risk-averse strategies. The joint interpretation, which unifies the two views of [BBC$^+$14] can be solved in polynomial time. For the conjunctive interpretation (and its combination with the joint one), we have provided an algorithm working in time polynomial in the size of MDP, but exponential in the number of different rewards. While this is not a severe limitation for practical purposes, the complexity of this problem remains an interesting open question.

References

[Alt99] E. Altman, *Constrained Markov Decision Processes (Stochastic Modeling)*. Chapman & Hall/CRC, 1999.

[BBC$^+$14] T. Brázdil, V. Brožek, K. Chatterjee, V. Forejt, and A. Kučera, "Two views on multiple mean-payoff objectives in markov decision processes," *Logical Methods in Computer Science*, vol. 10, no. 1, 2014. [Online]. Available: http://dx.doi.org/10.2168/LMCS-10(1:13)2014

[BFRR14] V. Bruyère, E. Filiot, M. Randour, and J. Raskin, "Meet your expectations with guarantees: Beyond worst-case synthesis in quantitative games," in *STACS'14*, 2014, pp. 199–213.

[BK08] C. Baier and J.-P. Katoen, *Principles of Model Checking*. MIT Press, 2008.

[CFW13] K. Chatterjee, V. Forejt, and D. Wojtczak, "Multi-objective discounted reward verification in graphs and mdps," in *LPAR'13*, 2013, pp. 228–242.

[Cha07] K. Chatterjee, "Markov decision processes with multiple long-run average objectives," in *FSTTCS*, ser. Lecture Notes in Computer Science, V. Arvind and S. Prasad, Eds., vol. 4855. Springer, 2007, pp. 473–484.

[CMH06] K. Chatterjee, R. Majumdar, and T. A. Henzinger, "Markov decision processes with multiple objectives," in *STACS*, ser. Lecture Notes in Computer Science, B. Durand and W. Thomas, Eds., vol. 3884. Springer, 2006, pp. 325–336.

[CY95] C. Courcoubetis and M. Yannakakis, "The complexity of probabilistic verification," *Journal of the ACM*, vol. 42, no. 4, pp. 857–907, 1995.

[CY98] ——, "Markov decision processes and regular events," *Automatic Control, IEEE Transactions on*, vol. 43, no. 10, pp. 1399–1418, Oct. 1998.

[dA97] L. de Alfaro, "Formal verification of probabilistic systems," Ph.D. dissertation, Stanford University, 1997.

[DEKM98] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge Univ. Press, 1998.

[EKVY08] K. Etessami, M. Kwiatkowska, M. Vardi, and M. Yannakakis, "Multi-objective model checking of Markov decision processes," *LMCS*, vol. 4, no. 4, pp. 1–21, 2008.

[FKN$^+$11] V. Forejt, M. Z. Kwiatkowska, G. Norman, D. Parker, and H. Qu, "Quantitative multi-objective verification for probabilistic systems," in *TACAS*, ser. Lecture Notes in Computer Science, P. A. Abdulla and K. R. M. Leino, Eds., vol. 6605. Springer, 2011, pp. 112–127.

[FKP12] V. Forejt, M. Z. Kwiatkowska, and D. Parker, "Pareto curves for probabilistic model checking," in *ATVA'12*, 2012, pp. 317–332.

[FV97] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*. Springer-Verlag, 1997.

[How60] H. Howard, *Dynamic Programming and Markov Processes*. MIT Press, 1960.

[Kar84] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1984, Washington, DC, USA*, R. A. DeMillo, Ed. ACM, 1984, pp. 302–311. [Online]. Available: http://doi.acm.org/10.1145/800057.808695

[KGFP09] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.

[KNP02] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM: Probabilistic symbolic model checker," in *TOOLS' 02*. LNCS 2324, Springer, 2002, pp. 200–204.

[Kos88] J. Koski, "Multicriteria truss optimization," in *Multicriteria Optimization in Engineering and in the Sciences*, W. Stadler, Ed. Plenum Press, 1988.

[Owe95] G. Owen, *Game Theory*. Academic Press, 1995.

[Put94] M. L. Puterman, *Markov Decision Processes*. J. Wiley and Sons, 1994.

[PY00] C. H. Papadimitriou and M. Yannakakis, "On the approximability of trade-offs and optimal access of web sources," in *FOCS*. IEEE Computer Society, 2000, pp. 86–92.

[Roy88] H. Royden, *Real analysis*, 3rd ed. Prentice Hall, 12 Feb. 1988.

[RRS14] M. Randour, J. Raskin, and O. Sankur, "Percentile queries in multi-dimensional markov decision processes," *CoRR*, vol. abs/1410.4801, 2014.

[SCK04] R. Szymanek, F. Catthoor, and K. Kuchcinski, "Time-energy design space exploration for multi-layer memory architectures," in *DATE*. IEEE Computer Society, 2004, pp. 318–323.

[Seg95] R. Segala, "Modeling and verification of randomized distributed real-time systems," Ph.D. dissertation, MIT, 1995, technical Report MIT/LCS/TR-676.

[Var85] M. Vardi, "Automatic verification of probabilistic concurrent finite state programs," in *Proc. FOCS'85*. IEEE Computer Society Press, 1985, pp. 327–338.

[WL99] C. Wu and Y. Lin, "Minimizing risk models in Markov decision processes with policies depending on target values," *Journal of Mathematical Analysis and Applications*, vol. 231, no. 1, pp. 47–67, 1999.

[YC03] P. Yang and F. Catthoor, "Pareto-optimization-based run-time task scheduling for embedded systems," in *CODES+ISSS*, R. Gupta, Y. Nakamura, A. Orailoglu, and P. H. Chou, Eds. ACM, 2003, pp. 120–125.

A Proof part 1: Witness strategy induces solution to L

Let ϱ be a strategy such that $\forall i \in [n]$

- $\mathbb{P}^\sigma[\mathrm{lr}_{\inf}(\boldsymbol{r})_i \geq \boldsymbol{sat}(i)] \geq \boldsymbol{pr}_i$
- $\mathbb{E}^\sigma[\mathrm{lr}_{\inf}(\boldsymbol{r})_i] \geq \boldsymbol{exp}_i$

We construct a solution to the system L.

A.1 Recurrent behaviour and Equations 4–7

We start with constructing values for variables x_a.

In general, the frequencies $\mathrm{freq}^\varrho(a)$ of the actions may not be well defined, because the defining limits may not exist. Further, it may be unavoidable to have different frequencies for several sets of runs of positive measure. There are two tricks to overcome this difficulty. Firstly, we partition the runs into several classes depending on which parts of the objective they achieve. Secondly, within each class we pick suitable values lying between $\mathrm{lr}_{\inf}(\boldsymbol{r})$ and $\mathrm{lr}_{\sup}(\boldsymbol{r})$ of these runs.

For $N \subseteq [n]$, let

$$\Omega_N = \{\omega \in \mathit{Runs} \mid \forall i \in N : \mathrm{lr}_{\inf}(\boldsymbol{r})(\omega)_i \geq \boldsymbol{sat}_i \wedge \forall i \notin N : \mathrm{lr}_{\inf}(\boldsymbol{r})(\omega)_i < \boldsymbol{sat}_i\}$$

Then Ω_N for $N \subseteq [n]$ form a paritioning of Runs. We define $f_N(a)$, lying between $\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{P}^\varrho[A_t = a \mid \Omega_N]$ and $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{P}^\varrho[A_t = a \mid \Omega_N]$, which can be safely substituted for $x_{a,N}$ in L. Since every infinite sequence contains an infinite convergent subsequence, there is an increasing sequence of indices, $T_0, T_1, \ldots$, such that the following limit exists for each action $a \in A$

$$f_N(a) \coloneqq \lim_{\ell\to\infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\varrho[A_t = a \mid \Omega_N] \cdot \mathbb{P}^\varrho[\Omega_N]$$

We set $x_{a,N} \coloneqq f_N(a)$ for all $a \in A$ and $N \subseteq [n]$ (where $x_{a,N} = 0$ whenever $\mathbb{P}^\varrho[\Omega_N] = 0$). We show that (in)equations 4–7 of L are satisfied.

Equation 4 For $t \in \mathbb{N}$, let

$$\Omega_{N|t} = \{\sigma[1]\cdots\sigma[t]\rho \mid \sigma \in \Omega_N\}$$

denote the (cylinders generated by) projection of Ω_N to first t steps. Then

$$\Omega_{N|1} \supseteq \Omega_{N|2} \supseteq \cdots \supseteq \Omega_N \tag{3}$$

$$\lim_{t\to\infty} \mathbb{P}^\varrho\big[\Omega_{N|t} \setminus \Omega_N\big] = 0 \tag{4}$$

For all $s \in S$ and $N \subseteq [n]$, we have

$$\sum_{a\in A} f_N(a) \cdot \delta(a)(s) = \sum_{a\in \mathit{Act}(s)} f_N(a)$$

trivialy for $\mathbb{P}^\varrho[\Omega_N] = 0$, and whenever $\mathbb{P}^\varrho[\Omega_N] > 0$ we have

$$
\begin{aligned}
&\frac{1}{\mathbb{P}^\varrho[\Omega_N]} \sum_{a \in A} f_N(a) \cdot \delta(a)(s) \\
&= \sum_{a \in A} \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\varrho[A_t = a \ \mid \Omega_N] \cdot \delta(a)(s) && \text{(definition of } f_N\text{)} \\
&= \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \sum_{a \in A} \mathbb{P}^\varrho[A_t = a \ \mid \Omega_N] \cdot \delta(a)(s) && \text{(linearity of the limit)} \\
&= \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \sum_{a \in A} \Big(\mathbb{P}^\varrho[A_t = a \mid \Omega_N] \cdot \frac{\mathbb{P}^\varrho[\Omega_N]}{\mathbb{P}^\varrho\big[\Omega_{N|t}\big]} + \\
&\qquad \mathbb{P}^\varrho\big[A_t = a \mid \Omega_{N|t} \setminus \Omega_N\big] \cdot \left(1 - \frac{\mathbb{P}^\varrho[\Omega_N]}{\mathbb{P}^\varrho\big[\Omega_{N|t}\big]}\right) \Big) \cdot \delta(a)(s) && \text{(by 4)} \\
&= \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \sum_{a \in A} \mathbb{P}^\varrho\big[A_t = a \ \mid \Omega_{N|t}\big] \cdot \delta(a)(s) && \text{(by 3)} \\
&= \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\varrho\big[S_{t+1} = s \ \mid \Omega_{N|t}\big] && \text{(definition of } \delta\text{)} \\
&= \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\varrho\big[S_{t+1} = s \mid \Omega_{N|t+1}\big] \cdot \frac{\mathbb{P}^\varrho\big[\Omega_{N|t+1}\big]}{\mathbb{P}^\varrho\big[\Omega_{N|t}\big]} + \\
&\qquad \mathbb{P}^\varrho\big[S_{t+1} = s \mid \Omega_{N|t} \setminus \Omega_{N|t+1}\big] \cdot \left(1 - \frac{\mathbb{P}^\varrho\big[\Omega_{N|t+1}\big]}{\mathbb{P}^\varrho\big[\Omega_{N|t}\big]}\right) && \text{(by 3)} \\
&= \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\varrho\big[S_{t+1} = s \ \mid \Omega_{N|t+1}\big] && \text{(by 4)} \\
&= \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\varrho\big[S_t = s \ \mid \Omega_{N|t}\big] && \text{(reindexing and Cesaro limit)} \\
&= \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \sum_{a \in Act(s)} \mathbb{P}^\varrho\big[A_t = a \ \mid \Omega_{N|t}\big] \quad (s \text{ must be followed by } a \in Act(s)) \\
&= \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \sum_{a \in Act(s)} \mathbb{P}^\varrho[A_t = a \ \mid \Omega_N] && \text{(as above by 3 and 4)} \\
&= \sum_{a \in Act(s)} \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\varrho[A_t = a \ \mid \Omega_N] && \text{(linearity of the limit)} \\
&= \frac{1}{\mathbb{P}^\varrho[\Omega_N]} \sum_{a \in Act(s)} f_N(a) \,. && \text{(definition of } f_N\text{)}
\end{aligned}
$$

Equation 5 For all $i \in [n]$, we have

$$\sum_{N \subseteq [n]} \sum_{a \in A} x_{a,N} \cdot \boldsymbol{r}_i(a) \;\geq\; \mathbb{E}^{\varrho}_{s_0}[\mathrm{lr}_{\inf}(\boldsymbol{r}_i)] \geq \boldsymbol{exp}_i \tag{5}$$

where the first inequality follows from:

$$\begin{aligned}
&\sum_{N \subseteq [n]} \sum_{a \in A} x_{a,N} \cdot \boldsymbol{r}_i(a) \\
&= \sum_{\substack{N \subseteq [n] \\ \mathbb{P}^{\varrho}[\Omega_N] > 0}} \sum_{a \in A} f_N(a) \cdot \boldsymbol{r}_i(a) && \text{(definition of } x_{a,N}\text{)} \\
&= \sum_{\substack{N \subseteq [n] \\ \mathbb{P}^{\varrho}[\Omega_N] > 0}} \sum_{a \in A} \boldsymbol{r}_i(a) \cdot \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^{\varrho}[A_t = a \mid \Omega_N] \cdot \mathbb{P}^{\varrho}[\Omega_N] && \text{(definition of } f_N\text{)} \\
&= \sum_{\substack{N \subseteq [n] \\ \mathbb{P}^{\varrho}[\Omega_N] > 0}} \mathbb{P}^{\varrho}[\Omega_N] \cdot \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \sum_{a \in A} \boldsymbol{r}_i(a) \cdot \mathbb{P}^{\varrho}[A_t = a \mid \Omega_N] \\
& && \text{(linearity of the limit)} \\
&\geq \sum_{\substack{N \subseteq [n] \\ \mathbb{P}^{\varrho}[\Omega_N] > 0}} \mathbb{P}^{\varrho}[\Omega_N] \cdot \liminf_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in A} \boldsymbol{r}_i(a) \cdot \mathbb{P}^{\varrho}[A_t = a \mid \Omega_N] \\
& && \text{(definition of lim inf)} \\
&= \sum_{\substack{N \subseteq [n] \\ \mathbb{P}^{\varrho}[\Omega_N] > 0}} \mathbb{P}^{\varrho}[\Omega_N] \cdot \liminf_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}^{\varrho}_{s_0}[\boldsymbol{r}_i(A_t) \mid \Omega_N] \\
& && \text{(definition of the expectation)} \\
&\geq \sum_{\substack{N \subseteq [n] \\ \mathbb{P}^{\varrho}[\Omega_N] > 0}} \mathbb{P}^{\varrho}[\Omega_N] \cdot \mathbb{E}^{\varrho}_{s_0}[\mathrm{lr}_{\inf}(\boldsymbol{r}_i) \mid \Omega_N] && \text{(Fatou's lemma)} \\
&= \mathbb{E}^{\varrho}_{s_0}[\mathrm{lr}_{\inf}(\boldsymbol{r}_i)] && (\Omega_N\text{'s partition } \mathsf{Runs})
\end{aligned}$$

Although Fatou's lemma (see, e.g. [Roy88, Chapter 4, Section 3]) requires the function $\boldsymbol{r}_i(A_t)$ be nonnegative, we can replace it with the nonnegative function $\boldsymbol{r}_i(A_t) - \min_{a \in A} \boldsymbol{r}_i(a)$ and add the subtracted constant afterwards.

Equation 6 For $C \in \mathsf{MEC}$, let

$$\Omega_C = \{\rho \in \mathsf{Runs} \mid \exists n_0 : \forall n > n_0 : \rho[n] \in C\}$$

denote the set of runs with a suffix in C. Since almost every run eventually remains in a MEC, e.g. [CY98, Proposition 3.1], $\{\Omega_C \mid C \in \mathsf{MEC}\}$ partitions

Runs. For the same reason, actions not in MECs are almost surely taken only finitely many times and thus

$$x_{a,N} = 0 \text{ for } a \notin \bigcup \mathsf{MEC}, N \subseteq [n] \tag{6}$$

For all $C \in \mathsf{MEC}, N \subseteq [n], i \in N$

$$\sum_{a \in C} x_{a,N} \cdot \boldsymbol{r}_i(a) \geq \sum_{a \in C} x_{a,N} \cdot \boldsymbol{sat}_i$$

follows trivialy for $\mathbb{P}^\varrho[\Omega_N] = 0$, and whenever $\mathbb{P}^\varrho[\Omega_N] > 0$ we have

$$\begin{aligned}
&\sum_{a \in C} x_{a,N} \cdot \boldsymbol{r}_i(a) \\
&\geq \liminf_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in C} \boldsymbol{r}_i(a) \cdot \mathbb{P}^\varrho[A_t = a \mid \Omega_N] \cdot \mathbb{P}^\varrho[\Omega_N] \\
&\qquad \text{(as above by def. of } x_{a,N}, f_N, \text{ linearity of lim, def. of lim inf)} \\
&= \liminf_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in C} \Big(\boldsymbol{r}_i(a) \cdot \mathbb{P}^\varrho[A_t = a \mid \Omega_N \cap \Omega_C] \cdot \frac{\mathbb{P}^\varrho[\Omega_N \cap \Omega_C]}{\mathbb{P}^\varrho[\Omega_N]} + \\
&\qquad \boldsymbol{r}_i(a) \cdot \mathbb{P}^\varrho[A_t = a \mid \Omega_N \setminus \Omega_C] \cdot \frac{\mathbb{P}^\varrho[\Omega_N \setminus \Omega_C]}{\mathbb{P}^\varrho[\Omega_N]} \Big) \cdot \mathbb{P}^\varrho[\Omega_N] \\
&\qquad \text{(partitioning of Runs)} \\
&= \liminf_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in C} \boldsymbol{r}_i(a) \cdot \mathbb{P}^\varrho[A_t = a \mid \Omega_N \cap \Omega_C] \cdot \mathbb{P}^\varrho[\Omega_N \cap \Omega_C] \\
&\qquad (\lim_{T \to \infty} \mathbb{P}^\varrho[A_t = a \mid \Omega_N \setminus \Omega_C] = 0 \text{ for } a \in C) \\
&\geq \mathbb{P}^\varrho[\Omega_N \cap \Omega_C] \cdot \mathbb{E}^\varrho_{s_0}[\mathrm{lr}_{\inf}(\boldsymbol{r}_i) \mid \Omega_N \cap \Omega_C] \\
&\qquad \text{(as above by def. of expectation and Fatou's lemma)} \\
&\geq \mathbb{P}^\varrho[\Omega_N \cap \Omega_C] \cdot \boldsymbol{sat}_i \qquad \text{(by definition of } \Omega_N \text{ and } i \in N)
\end{aligned}$$

It remains to prove the following:

Claim 5. For $N \subseteq [n]$ and $C \in \mathsf{MEC}$, we have $\sum_{a \in C} x_{a,N} = \mathbb{P}^\varrho[\Omega_N \cap \Omega_C]$.

Proof.

$$\begin{aligned}
&\sum_{a \in C} x_{a,N} \\
&= \liminf_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in C} \mathbb{P}^\varrho[A_t = a \mid \Omega_N \cap \Omega_C] \cdot \mathbb{P}^\varrho[\Omega_N \cap \Omega_C] \\
&\qquad \text{(as above with factor } \boldsymbol{r}_i(a) \text{ left out)} \\
&= \mathbb{P}^\varrho[\Omega_N \cap \Omega_C] \cdot \liminf_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in C} \mathbb{P}^\varrho[A_t = a \mid \Omega_N \cap \Omega_C] \\
&\qquad \text{(linearity of the limit)}
\end{aligned}$$

$$= \mathbb{P}^\varrho[\Omega_N \cap \Omega_C] \cdot \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{P}^\varrho[A_t \in C \mid \Omega_N \cap \Omega_C]$$

(taking two different action at time t are disjoint events)

$$= \mathbb{P}^\varrho[\Omega_N \cap \Omega_C] \qquad (A_t \in C \text{ for all but finitely many } t \text{ on } \Omega_C)$$

□

Equation 7 For every $i \in [n]$, by assumption on the strategy ϱ

$$\sum_{N\subseteq[n]:i\in N} \mathbb{P}^\varrho[\Omega_N] = \mathbb{P}^\varrho[\omega \in \mathsf{Runs} \mid \mathrm{lr}_{\inf}(\boldsymbol{r})_i(\omega) \geq \boldsymbol{sat}(i)] \geq \boldsymbol{pr}_i$$

and the first term equals

$$\begin{aligned}
\sum_{N\subseteq[n]:i\in N} \sum_{a\in A} x_{a,N} &= \sum_{N\subseteq[n]:i\in N} \sum_{C\in\mathsf{MEC}} \sum_{a\in C} x_{a,N} = && \text{(by (6))}\\
&= \sum_{N\subseteq[n]:i\in N} \sum_{C\in\mathsf{MEC}} \mathbb{P}^\varrho[\Omega_N \cap \Omega_C] && \text{(by Claim 5)}\\
&= \sum_{N\subseteq[n]:i\in N} \mathbb{P}^\varrho[\Omega_N] && (\Omega_C\text{'s partition } \mathsf{Runs})
\end{aligned}$$

A.2 Transient behaviour and Equations 1–3

Now we set the values for y_χ, $\chi \in A \cup (S \times 2^{[n]})$, and prove that they satisfy Equations 1–3 of L when the values $f_N(a)$ are assigned to $x_{a,N}$. One could obtain the values y_χ using the methods of [Put94, Theorem 9.3.8], which requires the machinery of deviation matrices. Instead, we can first simplify the behaviour of ϱ in the transient part to memoryless using [BBC+14] and then obtain y_χ directly, like in [EKVY08], as expected numbers of taking actions. To this end, for a state s we define $\Diamond s$ to be the set of runs that contain s.

Similarly to [BBC+14, Proposition 4.2 and 4.5], we modify the MDP G into another MDP $\bar{G}$ as follows: For each $s \in S, N \subseteq [n]$, we add a new absorbing state $f_{s,N}$. The only available action for $f_{s,N}$ leads back to $f_{s,N}$ with probability 1. We also add a new action, $a_{s,N}$, to every $s \in S$ for each $N \subseteq [n]$. The distribution associated with $a_{s,N}$ assigns probability 1 to $f_{s,N}$. Finally, we remove all unreachable states. The construction of [BBC+14] is the same but only a single value is used for N.

Claim 6. There is a strategy $\bar{\varrho}$ in $\bar{G}$ such that for every $C \in \mathsf{MEC}$ and $N \subseteq [n]$,

$$\sum_{s\in C} \mathbb{P}^{\bar{\varrho}}[\Diamond f_{s,N}] = \mathbb{P}^\varrho[\Omega_C \cap \Omega_N]$$

Proof. First, we consider an MDP G' created from G in the same way as $\bar{G}$, but instead of $f_{s,N}$ for each $s \in S, N \subseteq [n]$, we only have a single f_s; similarly for actions a_s. As in [BBC+14, Lemma 4.6], we obtain a strategy ϱ' in G' such that $\sum_{s \in C} \mathbb{P}^{\bar{\varrho}}[\Diamond f_s] = \mathbb{P}^{\varrho}[\Omega_C]$. We modify ϱ' into $\bar{\varrho}$. It behaves as ϱ', but instead of taking action a_s with probability p, we take each action $a_{s,N}$ with probability $p \cdot \frac{\mathbb{P}^{\varrho}[\Omega_C \cap \Omega_N]}{\mathbb{P}^{\varrho}[\Omega_C]}$. (For $\mathbb{P}^{\varrho}[\Omega_C] = 0$, it is defined arbitrarily.) Then

$$\sum_{s \in C} \mathbb{P}^{\bar{\varrho}}[\Diamond f_{s,N}] = \sum_{s \in C} \frac{\mathbb{P}^{\varrho}[\Omega_C \cap \Omega_N]}{\mathbb{P}^{\varrho}[\Omega_C]} \cdot \mathbb{P}^{\varrho'}[\Diamond f_s] = \mathbb{P}^{\varrho}[\Omega_C \cap \Omega_N]$$

□

By [EKVY08, Theorem 3.2], there are values y_χ satisfying the following:

- Equation 1 is satisfied. Further, summing up Equation 1 for each s yields Equation 2.
- $y_{s,N} \geq \sum_{s \in C} \mathbb{P}^{\bar{\varrho}}[\Diamond f_{s,N}]$. By Claim 6 for each $C \in \mathsf{MEC}$ we thus have

  $$\sum_{s \in C} y_{s,N} \geq \mathbb{P}^{\varrho}[\Omega_C \cap \Omega_N]$$

  and summing up over all C and N we have

  $$\sum_{N \subseteq [n]} \sum_{s \in S} y_{s,N} \geq \sum_{N \subseteq [n]} \mathbb{P}^{\varrho}[\Omega_N]$$

  where the first term is 1 by Equation 2, the second term is 1 by partitioning of Runs, hence they are actually equal and thus

  $$\sum_{s \in C} y_{s,N} = \mathbb{P}^{\varrho}[\Omega_C \cap \Omega_N] = \sum_{a \in C} x_{a,N}$$

  where the last equality follows by Claim 5, yielding Equation 3.
- There is a memoryless strategy $\hat{\varrho}$ such that $\mathbb{P}^{\hat{\varrho}}[\Diamond f_{s,N}] = \mathbb{P}^{\bar{\varrho}}[\Diamond f_{s,N}]$. The value y_a is the expected number of taking a by $\hat{\varrho}$ (for actions a preserved in $\bar{G}$) and $y_{s,N} = \mathbb{P}^{\hat{\varrho}}[\Diamond f_{s,N}]$. By [EKVY08, Lemma 3.3] all y_a and $y_{s,N}$ are indeed finite values.

B Proof part 2: Solution to L induces witness strategy

In this section, we prove that a solution to the system L induces a witness strategy in the MDP.

We use $\mathbb{P}^{\sigma}_s$ to denote $\mathbb{P}^{\sigma}$ corresponding to MDP where the initial state is set to s; similarly for freq^{σ}_s. Whenever we say "for almost all runs conforming to σ" we mean "for every initial state s and all runs, but some with $\mathbb{P}^{\sigma}_s$-measure 0".

B.1 Recurrent behaviour

We start with the recurrent part. To this end, we consider strongly connected MDP and Equation 4 only, for simplicity, with only one fixed N. The crucial observation we set out to prove is that even if the flow of Equation 4 is "disconnected" we may still play the actions with the exact frequencies $x_{a,N}$ on almost all runs.

Proposition 1. *In a strongly connected MDP, a non-negative solution $\bar{x}_a$ to Equation 4. induces a strategy such that almost all conforming runs ω satisfy*

$$\mathrm{freq}^{\omega}(a) = \bar{x}_a / \sum_a \bar{x}_a$$

Proof. Firstly, we construct a strategy for each "strongly connected" part of the solution $\bar{x}_a$. Secondly, we connect the parts and thus average the frequencies. This happens at a cost of small error used for transiting between the strongly connected parts. Thirdly, we eliminate this error as we let the transiting happen with mass vanishing over time.

Lemma 6. *A non-negative solution $\bar{x}_a$ to Equation 4. induces a memoryless strategy ζ such that for every BSCCs D of G^{ζ}, $a \in D \cap A$, and almost all conforming runs ω in D*

$$\mathrm{freq}^{\omega}(a) = \bar{x}_a / \sum_{a \in D \cap A} \bar{x}_a$$

Proof. By Lemma 4.3 of [BBC+14], we get a (memoryless) strategy ζ such that $\mathrm{freq}^{\zeta}_d(a) = \bar{x}_a / \sum_{a \in D \cap A} \bar{x}_a$ for any $d \in D \cap S$. Moreover, for any $d \in D$ by the ergodic theorem, $\mathrm{freq}^{\omega}(a)$ is the same for $\mathbb{P}^{\zeta}_d$-almost all runs ω, hence equal to $\mathrm{freq}^{\zeta}_d(a)$. □

Lemma 7. *In a strongly connected MDP, for every $\varepsilon > 0$ there is a strategy ξ_ε such that almost all conforming runs ω satisfy that* freq^{ω} *is positive and*

$$\mathrm{freq}^{\omega}(a) > \bar{x}_a / \sum_a \bar{x}_a - \varepsilon$$

Proof. We obtain ξ_ε by a suitable perturbation of the strategy ζ from previous lemma in such a way that all actions get positive probabilities and the frequencies of actions change only slightly, similarly as in [BBC+14, Proposition 5.1, Part 2]. There exists an arbitrarily small (strictly) positive solution x'_a of Equations 4. Indeed, it suffices to consider a strategy τ which always takes the uniform distribution over the actions in every state and then assign $\mathrm{freq}^{\tau}(a)/M$ to x_a for sufficiently large M. As the system of Equations 4. is linear and homogeneous, assigning $\bar{x}_a + x'_a$ to x_a also solves this system and Lemma 6 gives us a strategy

ξ_ε satisfying $\mathrm{freq}^{\xi_\varepsilon}(a) = (\bar{x}_a + x'_a)/X$ where $X = \sum_{a' \in A} \bar{x}_{a'} + x'_{a'}$. We may safely assume that $\sum_{a' \in A} x'_{a'} \leq \frac{\varepsilon \cdot (\sum_{a \in A} \bar{x}_a)^2}{\bar{x}_a - \varepsilon \cdot (\sum_{a \in A} \bar{x}_a)}$. Thus, we obtain

$$\mathrm{freq}^{\omega}(a) > \bar{x}_a / \sum_{s,a} \bar{x}_a - \varepsilon \tag{7}$$

by the following sequence of (in)equalities.

$$\begin{aligned}
&\mathrm{freq}^{\omega}(a) \\
&\quad = \frac{\bar{x}_a + x'_a}{\sum_{a' \in A} \bar{x}_{a'} + x'_{a'}} && \text{(by Lemma 6)} \\
&\quad > \frac{\bar{x}_a}{\sum_{a' \in A} \bar{x}_{a'} + x'_{a'}} && \text{(by } x'_a > 0\text{)} \\
&\quad \geq \frac{\bar{x}_a}{\sum_{a' \in A} \bar{x}_{a'} + \frac{\varepsilon \cdot (\sum_{a \in A} \bar{x}_a)^2}{\bar{x}_a - \varepsilon \cdot (\sum_{a \in A} \bar{x}_a)}} && \text{(by } \textstyle\sum_{a' \in A} x'_{a'} \leq \frac{\varepsilon \cdot (\sum_{a \in A} \bar{x}_a)^2}{\bar{x}_a - \varepsilon \cdot (\sum_{a \in A} \bar{x}_a)}\text{)} \\
&\quad = \frac{\bar{x}_a \cdot (\bar{x}_a - \varepsilon \cdot (\sum_{a \in A} \bar{x}_a))}{\sum_{a' \in A} \bar{x}_{a'} \cdot (\bar{x}_a - \varepsilon \cdot (\sum_{a \in A} \bar{x}_a)) + \varepsilon \cdot (\sum_{a \in A} \bar{x}_a)^2} && \text{(rearranging)} \\
&\quad = \frac{\bar{x}_a \cdot (\bar{x}_a - \varepsilon \cdot (\sum_{a \in A} \bar{x}_a))}{\sum_{a' \in A} \bar{x}_{a'} \cdot \bar{x}_a} && \text{(rearranging)} \\
&\quad = \frac{\bar{x}_a - \varepsilon \cdot (\sum_{a \in A} \bar{x}_a)}{\sum_{a' \in A} \bar{x}_{a'}} && \text{(rearranging)} \\
&\quad = \frac{\bar{x}_a}{\sum_{a' \in A} \bar{x}_{a'}} - \varepsilon && \text{(rearranging)}
\end{aligned}$$

Concerning the complexity of computing ξ_ε, note that the binary representation of every coefficient in L has only polynomial length. As $\bar{x}_a$'s are obtained as a solution of (a part of) L, standard results from linear programming imply that each $\bar{x}_a$ has a binary representation computable in polynomial time. The numbers x'_a are also obtained by solving a part of L and restricted by $\left|\sum_{a' \in A} x'_{a'}\right| \leq \frac{\varepsilon \cdot (\sum_{a \in A} \bar{x}_a)^2}{\bar{x}_a - \varepsilon \cdot (\sum_{a \in A} \bar{x}_a)}$ which allows to compute a binary representation of x'_a in polynomial time. The strategy ξ_ε assigns to each action only small arithmetic expressions over $\bar{x}_a$ and x'_a. Hence, ξ_ε is computable in polynomial time. □

Lemma 8. *In a strongly connected MDP, let ξ_i be a sequence of strategies with each freq^{ξ_i} constant for almost all conforming runs and positive, such that $\lim_{i \to \infty} \mathrm{freq}^{\xi_i}$ is well defined. Then there is a strategy ξ with*

$$\mathrm{freq}^{\xi} = \lim_{i \to \infty} \mathrm{freq}^{\xi_i}$$

which is constant on almost all runs.

Proof. The proof folows the computation of [BBC+14, Proposition 5.1, Part "Moreover"]. Given $a \in A$, let $I_a : A \to \{0, 1\}$ be a function given by $I_a(a) = 1$

and $I_a(b) = 0$ for all $b \neq a$. Let us consider instead of ξ_i its subsequence ξ_j such that $\mathbb{P}_s^{\xi_j}\left[\mathrm{lr}_{\inf}(I_a) \geq \mathrm{freq}^{\xi}(a) - 2^{-j-1}\right] = 1$. Existence of such a subsequence is ensured by $\mathrm{freq}^{\xi} = \lim_{i\to\infty} \mathrm{freq}^{\xi_i}$. Note that for every $j \in \mathbb{N}$ there is $\kappa_j \in \mathbb{N}$ such that for all $a \in A$ and $s \in S$ we get

$$\mathbb{P}_s^{\xi_j}\left[\inf_{T\geq\kappa_j} \frac{1}{T}\sum_{t=0}^{T} I_a(A_t) \geq \mathrm{freq}^{\xi}(a) - 2^{-j}\right] \geq 1 - 2^{-j}.$$

Now let us consider a sequence $n_0, n_1, \ldots$ of numbers where $n_j \geq \kappa_j$ and $\frac{\sum_{k<j} n_j}{n_j} \leq 2^{-j}$ and $\frac{\kappa_{j+1}}{n_j} \leq 2^{-j}$. We define ξ to behave as ξ_1 for the first n_1 steps, then as ξ_2 for the next n_2 steps, then as ξ_3 for the next n_3 steps, etc. In general, denoting by N_j the sum $\sum_{k<j} n_k$, the strategy ξ behaves as ξ_j between the N_j'th step (inclusive) and N_{j+1}'th step (non-inclusive).

Let us give some intuition behind ξ. The numbers in the sequence $n_0, n_1, \ldots$ grow rapidly so that after ξ_j is simulated for n_j steps, the part of the history when ξ_k for $k < j$ were simulated becomes relatively small and has only minor impact on the current average reward (this is ensured by the condition $\frac{\sum_{k<j} n_k}{n_j} \leq 2^{-j}$). This gives us that almost every run has infinitely many prefixes on which the average reward w.r.t. I_a is arbitrarily close to $\mathrm{freq}^{\xi}(a)$ infinitely often. To get that $\mathrm{freq}^{\xi}(a)$ is also the limit-average reward, one only needs to be careful when the strategy ξ ends behaving as ξ_j and starts behaving as ξ_{j+1}, because then up to the κ_{j+1} steps we have no guarantee that the average reward is close to $\mathrm{freq}^{\xi}(a)$. This part is taken care of by picking n_j so large that the contribution (to the average reward) of the n_j steps according to ξ_j prevails over fluctuations introduced by the first κ_{j+1} steps according to ξ_{j+1} (this is ensured by the condition $\frac{\kappa_{j+1}}{n_j} \leq 2^{-j}$).

Let us now prove the correctness of the definition of ξ formally. We prove that almost all runs ω of G^{ξ} satisfy

$$\liminf_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T} I_a(A_t(\omega)) \geq \lim_{j\to\infty} \mathrm{freq}^{\xi_j}(a).$$

Denote by E_j the set of all runs $\omega = s_0 a_0 s_1 a_1 \ldots$ of G^{ξ} such that for some $\kappa_j \leq d \leq n_j$ we have

$$\frac{1}{d}\sum_{j=N_j}^{N_j+d} I_a(a_k) \quad < \quad \lim_{j\to\infty} \mathrm{freq}^{\xi_j}(a) - 2^{-j}.$$

We have $\mathbb{P}^{\xi}[E_j] \leq 2^{-j}$ and thus $\sum_{j=1}^{\infty} \mathbb{P}^{\xi}[E_j] = \frac{1}{2} < \infty$. By Borel-Cantelli lemma [Roy88], almost surely only finitely many of E_j take place. Thus, almost every run $\omega = s_0 a_0 s_1 a_1 \ldots$ of G^{ξ} satisfies the following: there is ℓ such that for all $j \geq \ell$ and all $\kappa_j \leq d \leq n_i$ we have that

$$\frac{1}{d}\sum_{k=N_j}^{N_j+d} I_a(a_k) \quad \geq \quad \mathrm{freq}^{\xi}(a) - 2^{-j}.$$

Consider $T \in \mathbb{N}$ such that $N_j \leq T < N_{j+1}$ where $j > \ell$. We need the following inequality

$$\frac{1}{T}\sum_{t=0}^{T} I_a(a_t) \quad \geq \quad (\text{freq}^{\xi}(a) - 2^{-j})(1 - 2^{j-i}) \tag{8}$$

which can be proved as follows. First, note that

$$\frac{1}{T}\sum_{t=0}^{T} I_a(a_t) \quad \geq \quad \frac{1}{T}\sum_{t=N_{j-1}}^{N_j-1} I_a(a_t) + \frac{1}{T}\sum_{t=N_j}^{T} I_a(a_t)$$

and that

$$\frac{1}{T}\sum_{t=N_{j-1}}^{N_j-1} I_a(a_t) = \frac{1}{n_j}\sum_{t=N_{j-1}}^{N_j-1} I_a(a_t) \cdot \frac{n_j}{T} \geq (\text{freq}^{\xi}(a) - 2^{-j})\frac{n_j}{T}$$

which gives

$$\frac{1}{T}\sum_{t=0}^{T} I_a(a_t) \;\geq\; (\text{freq}^{\xi}(a) - 2^{-j})\frac{n_j}{T} \;+\; \frac{1}{T}\sum_{t=N_j}^{T} I_a(a_t). \tag{9}$$

Now, we distinguish two cases. First, if $T - N_i \leq \kappa_{j+1}$, then

$$\frac{n_j}{T} \geq \frac{n_j}{N_{j-1} + n_j + \kappa_{j+1}} = 1 - \frac{N_{j-1} + \kappa_{j+1}}{N_{j-1} + n_j + \kappa_{j+1}} \geq (1 - 2^{1-j})$$

and thus, by Equation (9),

$$\frac{1}{T}\sum_{t=0}^{T} I_a(a_t) \quad \geq \quad (\text{freq}^{\xi}(a) - 2^{-j})(1 - 2^{1-j}).$$

Second, if $T - N_i \geq \kappa_{j+1}$, then

$$\begin{aligned}
\frac{1}{T}\sum_{t=N_j+1}^{T} I_a(a_t) &= \frac{1}{T - N_j}\sum_{t=N_j+1}^{T} I_a(a_t) \cdot \frac{T - N_j}{T} \\
&\geq (\text{freq}^{\xi}(a) - 2^{-j-1})\left(1 - \frac{N_{j-1} + n_j}{T}\right) \\
&\geq (\text{freq}^{\xi}(a) - 2^{-j-1})\left(1 - 2^{-j} - \frac{n_j}{T}\right)
\end{aligned}$$

and thus, by Equation (9),

$$\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T} I_a(a_t) &\geq (\text{freq}^{\xi}(a) - 2^{-j})\frac{n_j}{T} + (\text{freq}^{\xi}(a) - 2^{-j-1})\left(1 - 2^{-j} - \frac{n_j}{T}\right) \\
&\geq (\text{freq}^{\xi}(a) - 2^{-j})\left(\frac{n_j}{T} + \left(1 - 2^{-j} - \frac{n_j}{T}\right)\right)
\end{aligned}$$

$$\geq (\mathrm{freq}^{\xi}(a) - 2^{-j})(1 - 2^{-j})$$

which finishes the proof of Equation (8).

Since the sum in Equation (8) converges to $\mathrm{freq}^{\xi}(a)$ as j (and thus also T) goes to ∞, we obtain

$$\liminf_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} I_a(a_t) \geq \mathrm{freq}^{\xi}(a).$$

□

The strategy of the proposition is now constructed by Lemma 8 taking ξ_i to be $\xi_{1/i}$ from Lemma 7. □

Now we know that strategies within an end component can be merged into a strategy with frequencies corresponding to the solution of Equation 4 for each fixed N. Note that it uses infinite memory, but only needs a counter and to know the current state. Let ξ_N denote this strategy.

The reward of ξ_N is almost surely

$$\mathrm{lr}(\boldsymbol{r})(\omega) = \sum_a \bar{x}_a \cdot \boldsymbol{r}(a) / \sum_a \bar{x}_a$$

When the MDP is not strongly connected, we obtain such ξ_N in each MEC C and the respective reward of almost all runs in C is thus

$$\mathbb{E}_C^{\xi_N}[\mathrm{lr}(\boldsymbol{r})] := \sum_{a\in C\cap A} \bar{x}_{a,N} \cdot \boldsymbol{r}(a) / \sum_{a\in C\cap A} \bar{x}_{a,N} \tag{10}$$

B.2 Transient behaviour

We now consider the transient part of the solution that plays ξ_N's with various probabilities.

Proposition 2. *Given $\{\xi_N\}$, a non-negative solution $\bar{y}_a, \bar{y}_{s,N}$ to equation 1. induces a strategy σ with*

$$\mathbb{P}^{\sigma}[\textit{switch to } \xi_N \textit{ in } s] = \bar{y}_{s,N}$$

Proof (Proof idea). Instead of switching in s_i to the recurrent behaviour ξ as in [BBC+14, Proposition 4.2, Step 1], we branch this decision and switch to ξ_N with proportion $p_N := \frac{\bar{y}_{s_i,N}}{\sum_{N\subseteq[n]} \bar{y}_{s_i,N}}$. In other words, instead of switching in s_i to ξ with probability $p := \frac{\bar{x}_{s_i}}{y_C - \sum_{j=1}^{i-1} \bar{x}_{s_j}}$ we switch to ξ_N with probability $p \cdot p_N$ (where x_{s_i}, y_C are summations over all N of the original ones). □

Proof. For every MEC C of G, we denote by y_C the number $\sum_{s\in C}\sum_{N\subseteq[n]}\bar{y}_{s,N}$. According to the Lemma 4.4. of [BBC+14] we have a stochastic-update strategy ζ which stays eventually in each MEC C with probability y_C.

Then the strategy $\bar{\sigma}$ works as follows. For a run initiated in s_0, the strategy $\bar{\sigma}$ plays according to ζ until a BSCC of G^ζ is reached. This means that every possible continuation of the path stays in the current MEC C of G. Assume that C has states $s_1,\dots,s_k$. We denote by $\bar{x}_{s,N}$ the sum $\sum_{a\in Act(s)}\bar{x}_{a,N}$. At this point, the strategy $\bar{\sigma}$ changes its behavior as follows: First, the strategy $\bar{\sigma}$ strives to reach s_1 with probability one. Upon reaching s_1, it chooses randomly with probability $\frac{\bar{x}_{s_1,N}}{y_C}$ to behave as ξ_N forever, or otherwise to follow on to s_2. If the strategy $\bar{\sigma}$ chooses to go on to s_2, it strives to reach s_2 with probability one. Upon reaching s_2, the strategy $\bar{\sigma}$ chooses (randomly, with probability $\frac{\bar{x}_{s_2,N}}{y_C-\sum_{N\subseteq[n]}\bar{x}_{s_1,N}}$) to behave as ξ_N forever, or to follow on to s_3, and so, till s_k. That is, the probability of switching to ξ_N in s_i is $\frac{\bar{x}_{s_i,N}}{y_C-\sum_{j=1}^{i-1}\sum_{N\subseteq[n]}\bar{x}_{s_j,N}}$.

Since ζ stays in a MEC C with probability y_C, the probability that the strategy $\bar{\sigma}$ switches to ξ_N in s_i is equal to $\bar{x}_{s_i,N}$.

Same as in [BBC+14] we can transform the part of $\bar{\sigma}$ before switching to ξ_N to a memoryless strategy and thus get strategy σ. □

Corollary 1. *Given $\{\xi_N\}$, a non-negative solution $\bar{y}_{s,N},\bar{x}_{a,N}$ to Equations 1 and 3 induces a strategy σ with*

$$\mathbb{P}^\sigma[\textit{switch to } \xi_N \textit{ in } C] = \sum_{a\in C\cap A}\bar{x}_{a,N}$$

for every MEC C.

Proposition 3. *A solution to the system L induces a witness strategy.*

Proof. Consider the strategy σ of Corollary 1. We evaluate the strategy σ as follows:

$$\begin{aligned}
\mathbb{E}^\sigma[\mathrm{lr}_{\inf}(\boldsymbol{r})] &= \sum_{N\subseteq[n]}\sum_{C\in MEC}\mathbb{P}^\sigma[\text{switch to } \xi_N \text{ in } C]\cdot\mathbb{E}_C^{\xi_N}[\mathrm{lr}(\boldsymbol{r})] \\
&\qquad\qquad \text{(by Equation 2, } \sum_{N\subseteq[n]}\mathbb{P}^\sigma[\text{switch to } \xi_N]=1\text{)} \\
&= \sum_{N\subseteq[n]}\sum_{C\in MEC}\Big(\sum_{a\in C\cap A}\bar{x}_{a,N}\Big)\cdot\mathbb{E}_C^{\xi_N}\mathrm{lr}(\boldsymbol{r}) \qquad \text{(by Corollary 1)} \\
&= \sum_{N\subseteq[n]}\sum_{C\in MEC}\Big(\sum_{a\in C\cap A}\bar{x}_{a,N}\Big)\cdot\Big(\sum_{a\in C\cap A}\bar{x}_{a,N}\cdot\boldsymbol{r}(a)/\sum_{a\in C\cap A}\bar{x}_{a,N}\Big) \\
&\qquad\qquad \text{(by (10))} \\
&= \sum_{N\subseteq[n]}\sum_{a\in A}\bar{x}_{a,N}\cdot\boldsymbol{r}(a) \\
&\geq \boldsymbol{exp} \qquad \text{(by Equation 5)}
\end{aligned}$$

and for each i

$$\begin{aligned}
\mathbb{P}^\sigma[\mathrm{lr}(\boldsymbol{r})_i \geq \boldsymbol{sat}(i)] &= \sum_{N\subseteq[n]} \sum_{C\in MEC} \mathbb{P}^\sigma[\text{switch to } \xi_N \text{ in } C] \cdot \mathbf{1}_{\mathbb{E}_C^{\xi_N}[\mathrm{lr}(\boldsymbol{r})_i]\geq \boldsymbol{sat}_i} \\
&\geq \sum_{i\in N\subseteq[n]} \sum_{C\in MEC} \mathbb{P}^\sigma[\text{switch to } \xi_N \text{ in } C] \cdot \mathbf{1}_{\mathbb{E}_C^{\xi_N}[\mathrm{lr}(\boldsymbol{r})_i]\geq \boldsymbol{sat}_i} \\
&= \sum_{i\in N\subseteq[n]} \sum_{C\in MEC} \mathbb{P}^\sigma[\text{switch to } \xi_N \text{ in } C] \qquad \text{(see below)} \\
&= \sum_{i\in N\subseteq[n]} \sum_{C\in MEC} \sum_{a\in C\cap A} \bar{x}_{a,N} \\
&= \sum_{i\in N\subseteq[n]} \sum_{a\in A} \bar{x}_{a,N} \\
&\geq \boldsymbol{pr}(i) \qquad \text{(by Equation 7)}
\end{aligned}$$

where for each i and $N \subseteq [n]$ with $i \in N$ and MEC C, we have

$$\begin{aligned}
\mathbb{E}_C^{\xi_N} \mathrm{lr}(\boldsymbol{r})_i &= \sum_{a\in C\cap A} \bar{x}_{a,N} \cdot \boldsymbol{r}(a) \Big/ \sum_{a\in C\cap A} \bar{x}_{a,N} \\
&\geq \sum_{a\in C\cap A} \bar{x}_{a,N} \cdot \boldsymbol{sat}_i \Big/ \sum_{a\in C\cap A} \bar{x}_{a,N} \qquad \text{(by Equation 6)} \\
&= \boldsymbol{sat}_i
\end{aligned}$$

□