Unifying two views on multiple mean-payoff objectives in Markov decision processes

Krishnendu Chatterjee, Zuzana Komarkova, Jan Kretinsky

https://doi.org/10.15479/AT:IST-2015-318-v2-1

Deposited at: 12 Dec 2018 11:54 ISSN: 2664-1690

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 and Zuzana Komarkova and Jan Kretinsky

Technical Report No. IST-2015-318-v2+1
Deposited at 23 Feb 2015 09:14
http://repository.ist.ac.at/327/1/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 exist 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 optimization with respect to both objectives at once, thus unifying the existing semantics. Precisely, the goal is to optimize the expectation while ensuring the satisfaction constraint. Our problem captures the notion of optimization with respect to strategies that are risk-averse (i.e., ensure certain probabilistic guarantee). Our main results are as follows: First, we present algorithms for the decision problems 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. Second, 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 behaviours are Markov decision processes (MDPs) [18, 24, 15]. 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 action 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 objectives in context of MDPs is the *limit-average (or long-run average or mean-payoff)* objective that assigns to every run the average of the rewards over the run.

Single vs multiple objectives. MDPs with single mean-payoff objectives have been widely studied (see, e.g., [24, 15]), with many applications ranging from computational biology to analysis of security protocols, randomized algorithms, or robot planning, to name a few [2, 21, 13, 20]. In verification of probabilistic systems, MDPs are widely used, for concurrent probabilistic systems [9, 30], probabilistic systems operating in open environments [28, 11], and applied in diverse domains [2, 21]. 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, which has also been applied in several problems such as dynamic power management [16].

Two views. There exist two views in the study of MDPs with mean-payoff objectives [3]. 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 the expectation semantics, such as in inventory control, planning, and performance evaluation [24, 15]. 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 mean-payoff value be at least sat with probability at least pr. In the case with 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 *conjunctive 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 sets of satisfying runs for each reward may even be disjoint here. The second interpretation (namely, the *joint interpretation*) requires the satisfaction semantics for all rewards at once. Precisely, it requires that, with probability at least pr, the mean-payoff value vector be 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 [3]. While the joint interpretation of satisfaction has already been introduced and studied in [3], here we consider also the conjunctive interpretation, which was not considered in [3].

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. 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 special 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 is $\boldsymbol{pr} = \mathbf{1}$ (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 and mono cases, the two interpretations (conjunctive and joint) of the satisfaction semantics coincide, whereas in the multi-quant problem (which is the most general problem) we consider both the conjunctive and the joint interpretations.

Motivation. The motivation to study the problem we consider is twofold. Firstly, it presents a unifying approach that combines the two existing semantics for MDPs. Secondly and more importantly, 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 [5], 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 non-negative with at least 95%. This is an instance of *mono-quant* with $pr = 0.95$, $sat = 0$.
– For more dimensions, consider the example [24, Problems 6.1, 8.17]. A vendor assigns to each customer either a low or a high rank. Further, there is a decision the vendor makes each year either to invest money into sending a catalogue to the customer or not. Depending on the rank and on receiving a catalogue, the customer spends different amounts for vendor's products and the rank can change. The aim is to maximize the expected profit provided the catalogue is almost surely sent with frequency at most f. This is an instance of *multi-qual.* Further, one can extend this example to only require that the catalogue frequency does not exceeded f with 95% probability, but 5% best customers may still receive catalogues very often (instance of *multi-quant*).
– The following is again an instance of *multi-quant.* A gratis service for downloading 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 the premium service, not only have we a higher expectation of $10Mbps$, but also 95% of the connections are guaranteed to run on at least $5Mbps$ (satisfaction constraint). 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. While satisfying the guarantee, we 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 [22] and in multi-criterion optimization and decision making in both economics and engineering [19, 32, 29].

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

- *Algorithmic complexity:* What is the complexity of deciding whether a given vector represents an achievable solution, and if the answer is yes, then compute a witness strategy?
- *Strategy complexity:* What type of strategies is sufficient (and necessary) for achievable solutions?
- *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:

- *Algorithmic complexity.* We present algorithms for deciding whether a given vector is an achievable solution and constructing a witness strategy. 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 conjunctive interpretation where it is exponential.
- *Strategy complexity.* It is known that for both expectation and satisfaction semantics with single reward, deterministic memoryless[1] strategies are sufficient [15, 4, 3]. 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 [3], where no MDP-dependent 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 bound for the satisfaction problem of [3]. A complete picture of the strategy complexity, and improvement over previous results is given in Table 1 and Remark 2 on p. 25.

[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 $\frac{1}{\varepsilon}$, for $\varepsilon > 0$.

In summary, we unify two existing semantics, present comprehensive results related to algorithmic and strategy complexities for the unifying semantics, and even improve results for the existing semantics.

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 [8, 14, 17, 3] 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 [3, 25] 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 conjunctive and joint interpretations are combined, and we can optimize any linear combination of expectations. Finally, we can also optimize the probabilistic guarantees while ensuring the required expectation. For greater detail, 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 making use of conditional probability methods.

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* [24, 1]. The attention in the study of constrained MDPs has been mainly focused on restricted classes of MDPs, such as unichain MDPs, where all states are visited infinitely often under any strategy. Such a restriction guarantees the existence of memoryless optimal strategies. The more general problem of MDPs with multiple mean-payoff objectives was first considered in [7] and a complete picture was presented in [3]. The expectation and satisfaction semantics was considered in [3], and our work unifies the two different semantics for MDPs. For general MDPs, [8, 6] studied multiple discounted reward functions. MDPs with multiple ω-regular specifications were studied in [14]. It was shown that the Pareto curve can be approximated in polynomial time in the size of MDP and exponential in the number of specifications; the algorithm reduces the problem to MDPs with multiple reachability specifications, which can be solved by multi-objective linear programming [23]. In [17], the results of [14] 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 [25]. However, [25] 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 [31], 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 instead as an adversary was considered in [5]. In [5] the problem was considered for single reward for mean-payoff and shortest path. In contrast, though inspired by [5], we consider risk aversion for multiple reward functions with probabilistic guarantee (instead of adversarial guarantee), which is natural for MDPs. Moreover, [5] generalizes mean-payoff games, for which 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 first item of our contributions).

2 Preliminaries

2.1 Basic definitions

We mostly follow the basic definition of [3] 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 $1 \leq i \leq k$. Further, $\mathbf{1}$ denotes $(1, \ldots, 1)$. 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 $\mathbb{P}(\mathsf{Cone}(\ell_1 \cdots \ell_k)) = \mu(\ell_1) \cdot \prod_{i=1}^{k-1} P(\ell_i)(\ell_{i+1})$.

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 in 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 an action a 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 $i \geq 1$, we have $a_i \in \mathit{Act}(s_i)$ and $\delta(a_i)(s_{i+1}) > 0$. 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 $\sigma : (SA)^* S \to$

$Dist(A)$ 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 slightly different (though equivalent—see [3, 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 $\sigma_u : A \times S \times \mathsf{M} \to Dist(\mathsf{M})$ and $\sigma_n : S \times \mathsf{M} \to Dist(A)$ are *memory update* and *next move* functions, respectively, and α is the 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, i.e. $\sigma_n(s, m) \in Dist(Act(s))$.

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 we 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 $f : \mathsf{Runs} \to \mathbb{R}$ 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 M, 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 $B \subseteq \bigcup_{t \in T} Act(t)$ 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 $\omega = s_1 a_1 \cdots a_{k-1} s_k$ 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 the subset ordering. Given an MDP, the set of MECs is denoted by MEC.

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 $\mathrm{lr}(\boldsymbol{r}) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{r}(A_t)$ may be undefined for some runs, we consider maximizing the respective pointwise limit inferior:

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

i.e. for each $i \in [n]$ and $\omega \in \mathsf{Runs}$, $\mathrm{lr}_{\mathrm{inf}}(\boldsymbol{r})(\omega)_i = \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{r}(A_t(\omega))_i$. Similarly, we could define $\mathrm{lr}_{\mathrm{sup}}(\boldsymbol{r}) = \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{r}(A_t)$. However, maximizing limit superior is less interesting, see [3]. Further, the respective minimizing problems can be solved by maximization with opposite rewards.

This paper is concerned with the following tasks:

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

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

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

for $pr \in [0,1]$. Further, we consider the following important 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 example). We illustrate **(multi-quant-conjunctive)** with an MDP of Fig. 2 with $n = 2$, rewards as depicted, and $\boldsymbol{exp} = (1.1, 0.5)$, $\boldsymbol{sat} =$

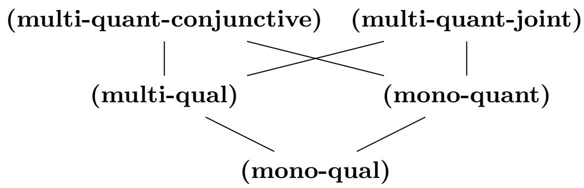


Fig. 1: Relationship of the defined problems with lower problems being specializations of the higher ones

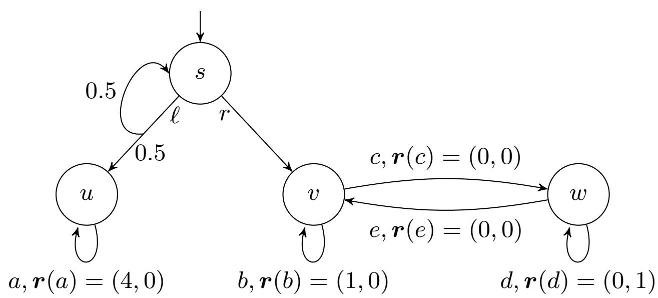


Fig. 2: An MDP with two-dimensional rewards

$(0.5, 0.5)$, $\boldsymbol{pr} = (0.8, 0.8)$. Observe that rewards of actions ℓ and r are irrelevant as these actions can almost surely be taken only finitely many times.

This instance is realizable and the witness strategy has the following properties. The strategy plays three "kinds" of runs. Firstly, due to $\boldsymbol{pr} = (0.8, 0.8)$, with probability 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, in order to ensure the expectation of the first reward, we reach the left MEC with probability 0.2 and play a. Thirdly, with probability 0.2 we reach again the right MEC but only play d with frequency 1, ensuring the expectation of the second reward.

In order to play these three kinds of runs, in the first step in s we take ℓ with probability 0.4 (arriving to u with probability 0.2) and r with probability 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 probability 0.25 we go to w and play the third kind of runs, and with probability 0.75 play the first kind of runs.

Observe that although both the expectation and satisfaction value thresholds for the second reward are 0.5, the only solution is not to play all runs with this reward, 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 at state w we have to play in two different ways, depending on which subset of value thresholds we intend to satisfy on each run. $\triangle$

3 Our solution

In this section, we briefly recall a solution to a previously considered problem and show our solution to the more general **(multi-quant-conjunctive)** realizability problem, along with an overview of the correctness proof. The solution to **(multi-quant-joint)** is derived and a detailed analysis of the special cases and the respective complexities is given in Section 5.

3.1 Previous results

In [3], a solution to the special case with only the (EXP) constraint has been given. The existence of a witness strategy was shown equivalent to the existence of a solution to 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 [3] for (EXP)

Intuitively, x_a is the expected frequency of using a on the long run; Equation 4 thus expresses the recurrent flow in MECs and Equation 5 the expected long-run average reward. However, before we can play according to x-variables, we have to reach MECs and switch from the transient behaviour to this recurrent behaviour. Equation 1 expresses the transient flow before switching. 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. To relate y- and x-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 [24]. However, the form with explicit y_s is more convenient for correctness proofs. 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, but again more convenient.

Further, apart from considering (EXP) separately, [3] 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 a single linear program for the **(multi-quant-conjunctive)** problem. A single subprogram for **(multi-quant-joint)** is derived in Section 5, thus unifying the previous 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. We first explain it on the qualitative case.

Solution to (multi-qual) When the additional constraint (SAT) is added so that almost all runs satisfy $\mathrm{lr}_{\mathrm{inf}}(\boldsymbol{r}) \geq \boldsymbol{sat}$, 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, that in a MEC, actions can be played on almost all runs with the given frequencies for any flow, see Corollary 1.

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

Solution to (multi-quant-conjunctive) 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 average 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.

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)**

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

Theorem 1. *Given a* ***(multi-quant-conjunctive)*** *realizability problem, the respective system L (in Fig. 4) 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 example). The linear program L for Example 1 is depicted in full in Appendix A. Here we spell out only some important points we need later: Equation 1 for state s

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

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

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

expresses that runs ending up in C and satisfying both satisfaction value 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 gain of all actions on runs that have committed to exceed the satisfaction value threshold either for the first reward, or for the first *and* the second reward. △

3.3 Proof overview

The first point The complexity follows immediately from the syntax of L and the existence of a polynomial-time algorithm for linear programming [27].

The second point Given a witness strategy σ, we construct values for variables so that a valid solution is obtained. The technical details can be found in Section 4.1 and Appendix C.

The proof of [3, Proposition 4.5], which inspires our proof, sets the values of x_a to be the expected frequency of using a by σ, i.e.

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

Since this Cesaro limit (expected frequency) 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 [3], 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 $\mathrm{lr}_{\mathrm{inf}}(\boldsymbol{r})_i \geq \boldsymbol{sat}_i$ for exactly all $i \in N$. N-good runs thus *jointly* satisfy the N-subset of the constraints. Now instead of using frequencies $f(a)$ of each action a, we use frequencies $f_N(a)$ of the action a on N-*good* runs separately, for each N. This requires some careful conditional probability considerations, in particular for Equations 1, 4, 6 and 7.

Example 3 (Running example). 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 threshold. Consequently, $x_{d,\{1,2\}} = 0.5 \cdot 0.6 = 0.3$ 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 taking actions during the "transient" behaviour 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 [14] 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 example). 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. Consequently, for instance for r, the expected number of taking this action 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" of runs (see Example 1). △

The third point Given a solution to L, we construct a witness strategy σ, which has a particular structure. The technical details can be found in Section 4.2 and Appendix D. The general pattern follows the proof method of [3, 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. Note that the flow described by x-variables has in general several disconnected components within the MEC, and thus actions connecting them must not be played with positive frequency. Yet there are 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 visits all the states of the MEC infinitely often, as opposed to strategies generated from the linear program in Fig. 3 in [3]. This simplifies the analysis and allows us to prove that stochastic update is not necessary for witness strategies. Therefore, deterministic update is sufficient, in particular also for (joint-SAT), which improves the strategy complexity known from [3].

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 example). 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 achieve this by playing actions c, e with probability $1/2^k$ in the kth step. In Section 4.2 we discuss the construction of the strategy from the solution in greater detail, necessary for later complexity discussion. $\triangle$

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 [3] 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 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, allowing us to maximize expectation while ensuring satisfaction.

Since our solution captures all the frequencies separately 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 a weighted reward expectation, i.e. $\boldsymbol{w} \cdot \mathrm{lr}_{\mathrm{inf}}(\boldsymbol{r})$, where $\boldsymbol{w}$ is the vector of weights for each reward dimension. Indeed, it can be expressed as the objective function $\boldsymbol{w} \cdot \sum_{a,N} x_{a,N} \cdot \boldsymbol{r}(a)$ of the linear program. Further, it is also relevant for the construction of the Pareto curve.
- We can also optimize satisfaction guarantees for given expectation thresholds. For more detail, see Section 5.3.
- We can easily express constraints for multiple as well as joint constraints $\mathbb{P}^\sigma\left[\bigwedge_{k_i} \mathrm{lr}_{\mathrm{inf}}(\boldsymbol{r}_{k_i}) \geq pr\right]$ 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 5. △

4 Technical proof of Theorem 1

Here we present the technical details of the constructions. Full proofs of the claims as well as further explanations can be found in Appendix C and Appendix D.

4.1 Item 2: Construction of solution to L

We show how a witness strategy σ induces a solution to L. For $N \subseteq [n]$, let

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

Then Ω_N, for $N \subseteq [n]$, form a partitioning of Runs. Since every infinite sequence contains an infinite convergent subsequence, there is an increasing sequence of indices, $T_0, T_1, \ldots$, such that, for each action $a \in A$, the following limit exists

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

whenever $\mathbb{P}^\sigma[\Omega_N] > 0$. In such a case, we set $x_{a,N} \coloneqq f_N(a)$, otherwise $x_{a,N} \coloneqq 0$. In Appendix C, we prove Equation 4–7 are then satisfied.

Moreover, we obtain the following claim necessary when relating the x-variables to the transient flow in Equation 3 later. For $C \in \mathsf{MEC}$, we denote the set of runs with a suffix in C by $\Omega_C = \{\rho \in \mathsf{Runs} \mid \exists n_0 : \forall n > n_0 : \rho[n] \in C\}$. Note that for every strategy, each run belongs to exactly one Ω_C almost surely.

Claim 1. For $N \subseteq [n]$ and $C \in \mathsf{MEC}$, we have

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

Now we set the values of y_χ for $\chi \in A \cup (S \times 2^{[n]})$. One could obtain the values y_χ using the methods of [24, Theorem 9.3.8], which requires the machinery of deviation matrices. Instead, we can first simplify the behaviour of σ in the transient part to memoryless using [3] and then obtain y_χ directly, like in [14], 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 [3, 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.

Claim 2. There is a strategy $\bar{\sigma}$ in $\bar{G}$ such that for every $C \in \mathsf{MEC}$ and $N \subseteq [n]$,

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

By [14, Theorem 3.2], there is a memoryless strategy $\bar{\sigma}$ satisfying the claim above such that $y_a := \sum_{t=1}^{\infty} \mathbb{P}^{\bar{\sigma}}[A_t = a]$ (for actions a preserved in $\bar{G}$) and $y_{s,N} := \mathbb{P}^{\bar{\sigma}}[\Diamond f_{s,N}]$ are finite values satisfying Equations 1, 2 and $y_{s,N} \geq \sum_{s\in C} \mathbb{P}^{\bar{\sigma}}[\Diamond f_{s,N}]$.

Therefore, by Claim 2 for each $C \in \mathsf{MEC}$, we obtain $\sum_{s\in C} y_{s,N} \geq \mathbb{P}^{\sigma}[\Omega_C \cap \Omega_N]$. This is actually an equality, by summing up over all C and N and using Equation 2. Now Equation 3 follows by Claim 1.

4.2 Item 3: Witness strategy induced by solution to L

Here we investigate the construction of the strategy from a solution to L in more detail. This will be necessary for establishing the complexity bounds in Section 5.

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. To formalize the frequency of an action a on a run, let $\mathbb{1}_a$ be the indicator function of a, i.e. $\mathbb{1}_a(a) = 1$ and $\mathbb{1}_a(b) = 0$ for $a \neq b \in A$. Then $\boldsymbol{freq}_a = \mathrm{lr}_{\inf}(\mathbb{1}_a)$ defines a vector random variable, indexed by $a \in A$. For the moment, we focus on strongly connected MDPs, i.e. the whole MDP is a MEC, and with $N \subseteq [n]$ fixed.

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 a small error used for transiting between the strongly connected parts.

Claim 3. In a strongly connected MDP, let $x_{a,N}, a \in A$ be a non-negative solution to Equation 4 of system L for a fixed $N \subseteq [n]$ and $\sum_{a\in A} x_{a,N} > 0$. For every $\varepsilon > 0$, there is a strategy ζ^{ε} such that for all $a \in A$ almost surely

$$\boldsymbol{freq}_a > \frac{x_{a,N}}{\sum_{a\in A} x_{a,N}} - \varepsilon \, .$$

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

Claim 4. In a strongly connected MDP, let ξ^i be a sequence of strategies, each with $\boldsymbol{freq} = \boldsymbol{f}^i$ almost surely, and such that $\lim_{i\to\infty} \boldsymbol{f}^i$ is well defined. Then there is a strategy ξ such that almost surely

$$\boldsymbol{freq} = \lim_{i\to\infty} \boldsymbol{f}^i \, .$$

Using Claim 4 with ξ^i being $\zeta^{1/i}$ from Claim 3 we obtain:

Corollary 1. *For a strongly connected MDP, let $x_{a,N}, a \in A$ be a non-negative solution to Equation 4 of system L for a fixed $N \subseteq [n]$ and $\sum_{a \in A} x_{a,N} > 0$. Then there is a strategy ξ_N such that for each $a \in A$ almost surely*

$$\boldsymbol{freq}_a = \frac{x_{a,N}}{\sum_{a \in A} x_{a,N}} .$$

Note that all actions and states in a MEC are visited infinitely often. This will be later useful for the strategy complexity analysis. Moreover, the long-run average reward is the same for almost all runs, which is a stronger property than in [3, Lemma 4.3], which does not hold for the induced strategy there. We need this property here in order to combine the satisfaction requirements. Finally, note that the strategy uses an infinite memory, but with a nice structure. It only needs a counter and to know the current state. Such strategies are also referred as Markov strategies [15].

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 an element that it starts playing exactly as ξ_N. We can stitch all ξ_N's together as follows:

Claim 5. Let $\xi_N, N \subseteq [n]$ be strategies. Then every non-negative solution $y_a, y_{s,N}, x_{a,N}, a \in A, s \in S, N \subseteq [n]$ to Equations 1 and 3 effectively induces a strategy σ such that for every MEC C

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

and σ is memoryless before the switch.

In Appendix D, we prove that σ of Claim 5, for $\xi_N, N \subseteq [n]$ of Corollary 1, is indeed a witness strategy.

5 Algorithmic and strategy complexity

5.1 Algorithmic complexity

In this section, we discuss the solutions to and complexity of all the introduced problems.

(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 algorithmic 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 4.2). 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 non-zero variables of L indexed by N satisfy $N = [n]$ or $N = \emptyset$. The remaining variables can be left out of the system, see Appendix E. Since there are now $\mathcal{O}(|G| \cdot n)$ variables, the problem as well as its special cases can be decided in polynomial time.

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.*

Results on other combinations Further, as discussed in Remark 1 and in greater detail in Appendix E, we show that if we add a (joint-SAT) constraint to **(multi-quant-conjunctive)**, the running time of the algorithm stays the same: polynomial in the size of the MDP and exponential in the dimension. We also prove (in Appendix E) the matching hardness result:

Theorem 3. *The realizability problem of conjunction of (joint-SAT) and (conjunctive-SAT) is NP-hard (even without the (EXP) constraint).*

Theorem 3 contrasts Theorem 2: while extension of (joint-SAT) with (EXP) can be solved in polynomial time, extending (joint-SAT) with (conjunctive-SAT) makes the problem NP-hard. Intuitively, adding (conjunctive-SAT) enforces us to consider the subsets of dimensions, and explains the exponential dependency on the number of dimensions in Theorem 1 (though our lower bound does not work for (conjunctive-SAT) with (EXP)).

5.2 Strategy complexity

First, we recall the structure of witness strategies generated from L in Section 4.2. In the first phase, a memoryless strategy is applied to reach MECs and switch to the recurrent strategies ξ_N. This switch is performed as a stochastic update, remembering the following two pieces of information: (1) the binary decision to stay in the current MEC C forever, and (2) the set $N \subseteq [n]$, such that almost 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 ζ_N^ε is played.

Similarly to the traditional setting with the expectation or the satisfaction semantics considered separately, the case with a *single reward* is simpler.

Lemma 1. *Deterministic memoryless strategies are sufficient for witness strategies for* ***(mono-qual)****.*

Proof. For each MEC, there is a value, which is the maximal long-run average 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 $\boldsymbol{sat}$. A witness strategy can be chosen to maximize the expected long-run average (single-dimensional) reward, and thus also to be deterministic and memoryless [24]. 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. □

Further, both for the expectation and the satisfaction semantics, deterministic memoryless strategies are sufficient with a single reward even for *quantitative* queries [15, 4]. In contrast, we show that both randomization and memory is necessary in our combined setting even for ε-witness 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 strategies would either never leave $\{s, a\}$ or would leave it eventually almost surely. Finally, the argument applies to ε-witness strategies, since the interval for p contains neither 0 nor 1 for sufficiently small ε. △

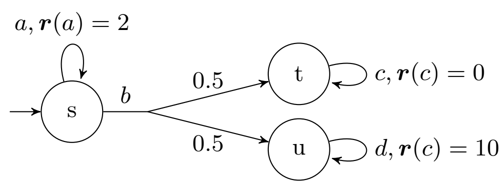


Fig. 5: An MDP with a single-dimensional reward, where both randomization and memory is necessary

In the rest of the section, we discuss bounds on the size of the memory and the degree of randomization. Due to [3, Section 5], infinite memory is indeed necessary for witnessing (joint-SAT) with $pr = 1$, hence also for **(multi-qual)**. However, as one of our main results, we prove that stochastic update at the moment of switching is not necessary, which improves also the result of [3].

Lemma 2. *Deterministic update is sufficient for witness strategies for* ***(multi-quant-conjuctive)*** *and* ***(multi-quant-joint)****. 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 average 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 the respective probability of leaving or staying in C.

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

Tossing-MEC case First, we assume that there are $toss, a, b \in C$ with $a, b \in Act(toss)$. Whenever σ should perform a step in $s \in C$ 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 consists of reaching *toss* with probability 1 (using a memoryless strategy), taking a, b with probabilities p_1, p_2, respectively, and making a deterministic update 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 as it already remembers the result of the (p_1, p_2)-toss, it changes the memory to m_1 or m_2 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 (e.g. various N), 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 implemented using finite memory. Indeed, since there are finitely many states in a MEC and σ is memoryless, there are only finitely many combinations of tosses to make and remember till the next simulated update of σ.

Tossfree-MEC case It remains to handle the case where, 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 ξ. 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

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

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. First, it 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

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

subsequently for each $i \in [\ell]$. After the last attempt with a_ℓ, if we are still in C, we update memory to stay in C forever (playing ξ).

Leaving C via a with probability *leave* can be done as follows. Let $\mathit{rate} = \sum_{s \notin C} \delta(a)(s)$ 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 $a \in \mathit{Act}(s)$ and play a with probability 1 and repeat this $\lfloor \mathit{leave}/\mathit{rate} \rfloor$-times and finally reach s once more and play a with probability $\mathit{leave}/\mathit{rate} - \lfloor \mathit{leave}/\mathit{rate} \rfloor$ 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 $\lceil 1/p_{\min} \rceil$, where $p_{\min}$ is the minimal positive probability in the MDP. □

Observe that the latter case of the proof is important for satisfaction objectives. It has not been considered in [3], where deterministic update is shown sufficient only for the expectation objective. Note that in general, conversion of a stochastic-update strategy to a deterministic-update strategy requires an infinite blow up in the memory [12].

As a consequence of Lemma 2, we obtain several bounds on memory size valid even for deterministic-update strategies. Firstly, infinite memory is only required for witness strategies:

Lemma 3. *Deterministic-update with finite memory is sufficient for ε-witness strategies for* ***(multi-quant-conjuctive)*** *and* ***(multi-quant-joint)****.*

Proof. After switching, memoryless strategies ζ_N^ϵ can be played instead of the sequence of $\zeta_N^{1/2^i}$ □

Secondly, infinite memory is only required for multiple rewards:

Lemma 4. *Deterministic-update strategies with finite memory are sufficient witness strategies for* ***(mono-quant)****.*

Proof. After switching in a MEC 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 average reward. □

Further, the construction in the toss-free case gives us a hint for the respective lower bound on memory, even for the single reward case.

Example 7. For deterministic-update ε-witness strategies for **(mono-quant)**, memory with size dependent on the transition probabilities is necessary. Indeed, consider the same realizability problem as in Example 6, but with a slightly modified MDP parametrized by λ, depicted in Fig. 6. Again, we have to remain in MEC $\{s, a\}$ with probability $p \in [0.1, 2/3]$. For ε-witness strategies the interval is slightly wider; let $\ell > 0$ denote the minimal probability with which any (ε-)witness strategy has to leave the MEC and all (ε-)witness strategies have to stay in the MEC with positive probability. We show that at least $\lceil \frac{\ell}{\lambda} \rceil$-memory is necessary. Observe that this setting also applies to the (EXP) setting of [3], e.g.

Table 1: Complexity results for each of the discussed cases. **U:** denotes upper bounds (which suffice for all MDPs) and **L:** lower bounds (which are required in general for some MDPs). Results without reference are induced by the specialization/generalization relation depicted in Fig. 1. The abbreviations stoch.-up., det.-up., rand., det., inf., fin., X-mem. stand for stochastic update, deterministic update, randomizing, deterministic, infinite-, finite- and X-memory strategies, respectively. Here n is the dimension of rewards and $p = 1/p_{\min}$ where $p_{\min}$ is the smallest positive probability in the MDP. Note that inf. actually means infinite memory consisting of a counter together with finite memory, see Section 4.2.

Case	Algorithmic complexity	Witness strategy complexity	ε-witness strategy complexity
(multi-quant-conj.)	$poly(\lvert G\rvert, 2^n)$ [Thm.1]	**U:** det.-up. [Lem.2] inf. **L:** rand. inf.	**U:** det.-up. fin.[Lem.3] **L:** rand. n-mem., for det.-up. p-mem.
(multi-quant-joint)	$poly(\lvert G\rvert, n)$ [Thm.2]	**U:** det.-up. inf. **L:** rand. inf.	**U:** det.-up. fin. **L:** rand. n-mem., for det.-up. p-mem.
(multi-qual)	$poly(\lvert G\rvert, n)$	**U:** det.-up. inf. **L:** rand. inf. [3, Sec.5]	**U:** det.-up. fin. **L:** rand. n-mem. [Ex.8]
(mono-quant)	$poly(\lvert G\rvert)$	**U:** stoch.-up. 2-mem. [Lem.5], det.-up. fin. [Lem.4] **L:** rand. mem., for det.-up. p-mem.	**U:** stoch.-up. 2-mem., det.-up. fin. **L:** rand. [Ex.6] mem. [Ex.6], for det.-up. p-mem. [Ex.7]
(mono-qual)	$poly(\lvert G\rvert)$	**U:** (trivially also **L:**) det. memoryless [Lem.1]	**U:** (trivially also **L:**) det. memoryless

exp $= (0.5, 0.5)$ and the MDP of Fig. 7. Therefore, we provide a lower bound also for this simpler case (no MDP-dependent lower bound is provided in [3]).

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

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

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

Although we have shown that stochastic update is not necessary, it may be helpful when memory is small.

Lemma 5. *Stochastic-update 2-memory strategies are sufficient for witness strategies for **(mono-quant)**.*

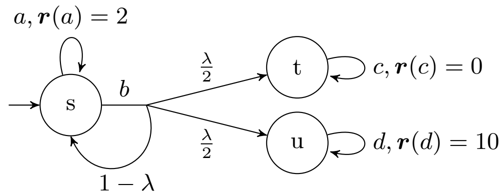


Fig. 6: An MDP family with a single-dimensional reward, where memory with size dependent on transition probabilities is necessary for deterministic-update strategies

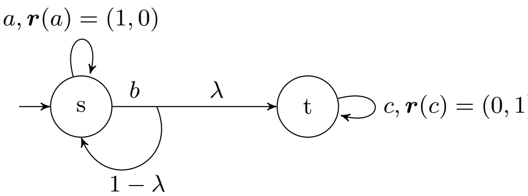


Fig. 7: An MDP family, where memory with size dependent on transition probabilities is necessary for deterministic-update strategies even for (EXP) studied in [3]

Proof. The strategy σ of Section 4.2, which reaches the MECs and stays in them with given probability, is memoryless up to the point of switch by Claim 5. Further, we can achieve the optimal value in each MEC using a memoryless strategy as in Lemma 4. $\square$

However, even with stochastic update, the sizes of the finite memories cannot be bounded by a constant for multiple rewards.

Example 8. Even ε-witness strategies may require 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. 8 illustrates the case with $n = 3$. Further, let $\boldsymbol{sat} = \mathbf{1}$ and $\boldsymbol{pr} = 1/n \cdot \mathbf{1}$.

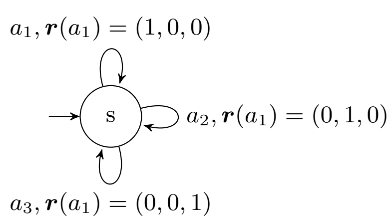


Fig. 8: An MDP where n-memory is necessary, depicted for $n = 3$

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 repetitively taken at s, showing the memory requirement. △

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

Theorem 4. *The bounds on the complexity of the witness and ε-witness strategies are as shown in Table 1.*

Remark 2. Along with our results for the new unifying semantics we present two improvements over the results for the existing semantics; one related to upper bounds and the other related to lower bounds:

1. *Upper bound.* While [3] shows that deterministic-update strategies are sufficient for (EXP), it does not present any such result for (joint-SAT), although both are special cases of **(multi-quant-joint)**. The result for (EXP) was simpler since the tossfree-MEC case of Lemma 2 is not relevant for (EXP), where the MEC is either almost surely left or almost surely stayed at. Thus while the proof of [3] for (joint-SAT) constructs witness strategies that require stochastic update, we improve this upper bound by showing deterministic-update strategies are sufficient for (joint-SAT), and in fact, even for **(multi-quant-joint)**.
2. *Lower bound.* Further, [3] shows that pure memoryless strategies are not sufficient for (EXP), thus establishing an MDP-independent lower bound of 2 for memory. For stochastic-update strategies, memory of size 2 suffices [3] and the bound is tight. However, for deterministic-update strategies, the bound is not tight, and our Example 7 shows that memory of size dependent on the MDP is necessary for (EXP). Thus our result improves the lower bound for deterministic-update strategies for (EXP). △

5.3 Pareto curve approximation

For a single reward, no Pareto curve is required and we can compute the optimal value of expectation in polynomial time by the linear program L with the objective function $\max \sum_{a\in A}(x_{a,\emptyset} + x_{a,\{1\}}) \cdot \boldsymbol{r}(a)$. For multiple rewards we obtain the following:

Theorem 5. *For $\varepsilon > 0$, an ε-approximation of the Pareto curve for **(multi-quant-conjunctive)** and **(multi-quant-joint)** can be constructed in time polynomial in $|G|$ and $\frac{1}{\varepsilon}$ and exponential in 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 [23], we can ε-approximate the Pareto curve in time polynomial in the size of the program and $\frac{1}{\varepsilon}$, and exponential in the number of objectives (dimension of $\boldsymbol{v}$). □

While applied to expectation, observe this method also applies to Pareto-curve approximation for satisfaction, in the same way; for more detail, see Appendix E.

6 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. We show that the joint interpretation can be solved in polynomial time. For both the conjunctive interpretation and its combination with the joint one, we present an algorithm that works in time polynomial in the size of MDP, but exponential in the number of rewards. We also prove that the latter problem is NP-hard. While the exponential in number of reward functions is not a limitation for practical purposes (as in most cases the number of reward functions is constant), the complexity of the conjunctive interpretation alone remains an interesting open question. Moreover, our algorithms for Pareto-curve approximation work in time polynomial in the size of MDPs and exponential in the number of reward functions. However, note that even for the special case of expectation semantics the current best known algorithms depend exponentially on the number of reward functions [3].

References

1. E. Altman, *Constrained Markov Decision Processes (Stochastic Modeling)*. Chapman & Hall/CRC, 1999.
2. C. Baier and J.-P. Katoen, *Principles of Model Checking*. MIT Press, 2008.
3. T. Brázdil, V. Brožek, K. Chatterjee, V. Forejt, and A. Kučera, "Two views on multiple mean-payoff objectives in markov decision processes," *LMCS*, vol. 10, no. 1, 2014.
4. T. Brázdil, V. Brožek, and K. Etessami, "One-counter stochastic games," in *FSTTCS*, ser. LIPIcs, K. Lodaya and M. Mahajan, Eds., vol. 8. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010, pp. 108–119.
5. 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.
6. K. Chatterjee, V. Forejt, and D. Wojtczak, "Multi-objective discounted reward verification in graphs and mdps," in *LPAR'13*, 2013, pp. 228–242.
7. K. Chatterjee, "Markov decision processes with multiple long-run average objectives," in *FSTTCS*, 2007, pp. 473–484.
8. K. Chatterjee, R. Majumdar, and T. A. Henzinger, "Markov decision processes with multiple objectives," in *STACS*, 2006, pp. 325–336.
9. C. Courcoubetis and M. Yannakakis, "The complexity of probabilistic verification," *Journal of the ACM*, vol. 42, no. 4, pp. 857–907, 1995.
10. ——, "Markov decision processes and regular events," *Automatic Control, IEEE Transactions on*, vol. 43, no. 10, pp. 1399–1418, Oct. 1998.
11. L. de Alfaro, "Formal verification of probabilistic systems," Ph.D. dissertation, Stanford University, 1997.
12. L. de Alfaro, T. A. Henzinger, and O. Kupferman, "Concurrent reachability games," *Theor. Comput. Sci*, vol. 386, no. 3, pp. 188–217, 2007.
13. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge Univ. Press, 1998.

14. 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.
15. J. Filar and K. Vrieze, *Competitive Markov Decision Processes*. Springer-Verlag, 1997.
16. V. Forejt, M. Z. Kwiatkowska, and D. Parker, "Pareto curves for probabilistic model checking," in *ATVA'12*, 2012, pp. 317–332.
17. V. Forejt, M. Z. Kwiatkowska, G. Norman, D. Parker, and H. Qu, "Quantitative multi-objective verification for probabilistic systems," in *TACAS*, 2011, pp. 112–127.
18. H. Howard, *Dynamic Programming and Markov Processes*. MIT Press, 1960.
19. J. Koski, "Multicriteria truss optimization," in *Multicriteria Optimization in Engineering and in the Sciences*, 1988.
20. 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.
21. M. Kwiatkowska, G. Norman, and D. Parker, "PRISM: Probabilistic symbolic model checker," in *TOOLS' 02*, 2002, pp. 200–204.
22. G. Owen, *Game Theory*. Academic Press, 1995.
23. C. H. Papadimitriou and M. Yannakakis, "On the approximability of trade-offs and optimal access of web sources," in *FOCS*, 2000, pp. 86–92.
24. M. L. Puterman, *Markov Decision Processes*. J. Wiley and Sons, 1994.
25. M. Randour, J. Raskin, and O. Sankur, "Percentile queries in multi-dimensional Markov decision processes," *CoRR*, vol. 1410.4801, 2014.
26. H. Royden, *Real analysis*, 3rd ed. Prentice Hall, 12 Feb. 1988.
27. A. Schrijver, *Theory of Linear and Integer Programming*. Wiley-Interscience, 1986.
28. R. Segala, "Modeling and verification of randomized distributed real-time systems," Ph.D. dissertation, MIT, 1995.
29. R. Szymanek, F. Catthoor, and K. Kuchcinski, "Time-energy design space exploration for multi-layer memory architectures," in *DATE*, 2004, pp. 318–323.
30. M. Vardi, "Automatic verification of probabilistic concurrent finite state programs," in *FOCS*, 1985, pp. 327–338.
31. 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.
32. P. Yang and F. Catthoor, "Pareto-optimization-based run-time task scheduling for embedded systems," in *CODES+ISSS*, 2003, pp. 120–125.

A Linear program $\boldsymbol{L}$ for Example 1

1. $1 + 0.5y_\ell = y_\ell + y_r + y_{s,\emptyset} + y_{s,\{1\}} + y_{s,\{2\}} + y_{s,\{1,2\}}$
   $0.5y_\ell + y_a = y_a + y_{u,\emptyset} + y_{u,\{1\}} + y_{u,\{2\}} + y_{u,\{1,2\}}$
   $y_r + y_b + y_e = y_b + y_c + y_{v,\emptyset} + y_{v,\{1\}} + y_{v,\{2\}} + y_{v,\{1,2\}}$
   $y_c + y_d = y_d + y_e + y_{w,\emptyset} + y_{w,\{1\}} + y_{w,\{2\}} + y_{w,\{1,2\}}$

2. $y_{u,\emptyset} + y_{u,\{1\}} + y_{u,\{2\}} + y_{u,\{1,2\}} + y_{v,\emptyset} + y_{v,\{1\}} + y_{v,\{2\}} + y_{v,\{1,2\}} + y_{w,\emptyset} + y_{w,\{1\}} + y_{w,\{2\}} + y_{w,\{1,2\}} = 1$

3. $y_{u,\emptyset} = x_{a,\emptyset}$
   $y_{u,\{1\}} = x_{a,\{1\}}$
   $y_{u,\{2\}} = x_{a,\{2\}}$
   $y_{u,\{1,2\}} = x_{a,\{1,2\}}$

   $y_{v,\emptyset} + y_{w,\emptyset} = x_{b,\emptyset} + x_{c,\emptyset} + x_{d,\emptyset} + x_{e,\emptyset}$
   $y_{v,\{1\}} + y_{w,\{1\}} = x_{b,\{1\}} + x_{c,\{1\}} + x_{d,\{1\}} + x_{e,\{1\}}$
   $y_{v,\{2\}} + y_{w,\{2\}} = x_{b,\{2\}} + x_{c,\{2\}} + x_{d,\{2\}} + x_{e,\{2\}}$
   $y_{v,\{1,2\}} + y_{w,\{1,2\}} = x_{b,\{1,2\}} + x_{c,\{1,2\}} + x_{d,\{1,2\}} + x_{e,\{1,2\}}$

4. $0.5x_{\ell,\emptyset} = x_{\ell,\emptyset} + x_{r,\emptyset}$
   $0.5x_{\ell,\{1\}} = x_{\ell,\{1\}} + x_{r,\{1\}}$
   $0.5x_{\ell,\{2\}} = x_{\ell,\{2\}} + x_{r,\{2\}}$
   $0.5x_{\ell,\{1,2\}} = x_{\ell,\{1,2\}} + x_{r,\{1,2\}}$

   $0.5x_{\ell,\emptyset} + x_{a,\emptyset} = x_{a,\emptyset}$
   $0.5x_{\ell,\{1\}} + x_{a,\{1\}} = x_{a,\{1\}}$
   $0.5x_{\ell,\{2\}} + x_{a,\{2\}} = x_{a,\{2\}}$
   $0.5x_{\ell,\{1,2\}} + x_{a,\{1,2\}} = x_{a,\{1,2\}}$

   $x_{r,\emptyset} + x_{b,\emptyset} + x_{e,\emptyset} = x_{b,\emptyset} + x_{c,\emptyset}$
   $x_{r,\{1\}} + x_{b,\{1\}} + x_{e,\{1\}} = x_{b,\{1\}} + x_{c,\{1\}}$
   $x_{r,\{2\}} + x_{b,\{2\}} + x_{e,\{2\}} = x_{b,\{2\}} + x_{c,\{2\}}$
   $x_{r,\{1,2\}} + x_{b,\{1,2\}} + x_{e,\{1,2\}} = x_{b,\{1,2\}} + x_{c,\{1,2\}}$

   $x_{c,\emptyset} + x_{d,\emptyset} = x_{d,\emptyset} + x_{e,\emptyset}$
   $x_{c,\{1\}} + x_{d,\{1\}} = x_{d,\{1\}} + x_{e,\{1\}}$
   $x_{c,\{2\}} + x_{d,\{2\}} = x_{d,\{2\}} + x_{e,\{2\}}$
   $x_{c,\{1,2\}} + x_{d,\{1,2\}} = x_{d,\{1,2\}} + x_{e,\{1,2\}}$

5. $\boldsymbol{r}(\ell)x_{\ell,\emptyset} + \boldsymbol{r}(\ell)x_{\ell,\{1\}} + \boldsymbol{r}(\ell)x_{\ell,\{2\}} + \boldsymbol{r}(\ell)x_{\ell,\{1,2\}} + \boldsymbol{r}(r)x_{r,\emptyset} + \boldsymbol{r}(r)x_{r,\{1\}} + \boldsymbol{r}(r)x_{r,\{2\}} + \boldsymbol{r}(r)x_{r,\{1,2\}} + (4,0)x_{a,\emptyset} + (4,0)x_{a,\{1\}} + (4,0)x_{a,\{2\}} + (4,0)x_{a,\{1,2\}} + (1,0)x_{b,\emptyset} + (1,0)x_{b,\{1\}} + (1,0)x_{b,\{2\}} + (1,0)x_{b,\{1,2\}} + (0,0)x_{c,\emptyset} + (0,0)x_{c,\{1\}} + (0,0)x_{c,\{2\}} + (0,0)x_{c,\{1,2\}} + (0,1)x_{d,\emptyset} + (0,1)x_{d,\{1\}} + (0,1)x_{d,\{2\}} + (0,1)x_{d,\{1,2\}} + (0,0)x_{e,\emptyset} + (0,0)x_{e,\{1\}} + (0,0)x_{e,\{2\}} + (0,0)x_{e,\{1,2\}} \geq (1.1, 0.5)$

6. $4x_{a,\{1\}} \geq 0.5x_{a,\{1\}}$
   $0 \geq 0.5x_{a,\{2\}}$

$4x_{a,\{1,2\}} \geq 0.5x_{a,\{1,2\}}$
$0 \geq 0.5x_{a,\{1,2\}}$

$x_{b,\{1\}} \geq 0.5x_{b,\{1\}} + 0.5x_{c,\{1\}} + 0.5x_{d,\{1\}} + 0.5x_{e,\{1\}}$
$x_{d,\{2\}} \geq 0.5x_{b,\{2\}} + 0.5x_{c,\{2\}} + 0.5x_{d,\{2\}} + 0.5x_{e,\{2\}}$
$x_{b,\{1,2\}} \geq 0.5x_{b,\{1,2\}} + 0.5x_{c,\{1,2\}} + 0.5x_{d,\{1,2\}} + 0.5x_{e,\{1,2\}}$
$x_{d,\{1,2\}} \geq 0.5x_{b,\{1,2\}} + 0.5x_{c,\{1,2\}} + 0.5x_{d,\{1,2\}} + 0.5x_{e,\{1,2\}}$

7. $x_{\ell,\{1\}} + x_{\ell,\{1,2\}} + x_{r,\{1\}} + x_{r,\{1,2\}} + x_{a,\{1\}} + x_{a,\{1,2\}} + x_{b,\{1\}} + x_{b,\{1,2\}} + x_{c,\{1\}} + x_{c,\{1,2\}} + x_{d,\{1\}} + x_{d,\{1,2\}} + x_{e,\{1\}} + x_{e,\{1,2\}} \geq 0.8$

   $x_{\ell,\{2\}} + x_{\ell,\{1,2\}} + x_{r,\{2\}} + x_{r,\{1,2\}} + x_{a,\{2\}} + x_{a,\{1,2\}} + x_{b,\{2\}} + x_{b,\{1,2\}} + x_{c,\{2\}} + x_{c,\{1,2\}} + x_{d,\{2\}} + x_{d,\{1,2\}} + x_{e,\{2\}} + x_{e,\{1,2\}} \geq 0.8$

Note that by Lemma 7 of Appendix D, variables $x_{\ell,N}, x_{r,N}$ for any $N \subseteq [n]$ can be omitted from the system as they are zero for any solution. Intuitively, transient actions cannot be used in the recurrent flows.

B Additional definitions

The proofs in Appendix use the following additional notation:

- For a run $\omega = \ell_1\ell_2\cdots$ and $n \in \mathbb{N}$, we denote the n-th location on the run by $\omega[n]$.
- When the initial state s is not clear from the context, we use $\mathbb{P}^\sigma_s$ to denote $\mathbb{P}^\sigma$ corresponding to the MDP where the initial state is set to s.
- 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 $\{s, a \mid (s, m, a) \in W\}$ of G, and we sometimes do not strictly distinguish between W and its associated end component.
- For $C \in \mathsf{MEC}$, let

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

  denote the set of runs with a suffix in C. Similarly, we define Ω_D for a BSCC D.
  Since almost every run eventually remains in a MEC, e.g. [10, Proposition 3.1], $\{\Omega_C \mid C \in \mathsf{MEC}\}$ "partitions" almost all runs. More precisely, for every strategy, each run belongs to exactly one Ω_C almost surely; i.e. a run never belongs to two Ω_C's and for every σ, we have $\mathbb{P}^\sigma\left[\bigcup_{C\in\mathsf{MEC}} \Omega_C\right] = 1$. Therefore, actions that are not in any MEC are almost surely taken only finitely many times.

C Proof of Theorem 1, item 2: Witness strategy induces solution to $\boldsymbol{L}$

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

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

We construct a solution to the system L. The proof method roughly follows that of [3, Proposition 4.5]. However, separate flows for "N-good" runs require some careful conditional probability considerations, in particular for Equations 4, 6 and 7.

C.1 Recurrent behaviour and Equations 4–7

We start with constructing values for variables $x_{a,N}, a \in A, N \subseteq [n]$.

In general, the frequencies $\mathrm{freq}^{\sigma}(a)$ 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 $\mathrm{lr}_{\mathrm{inf}}(\boldsymbol{r})$ and $\mathrm{lr}_{\mathrm{sup}}(\boldsymbol{r})$ of these runs.

For $N \subseteq [n]$, let

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

Then Ω_N, $N \subseteq [n]$ form a partitioning of Runs. Further, observe that runs of Ω_N are the runs where joint satisfaction holds, for all rewards $i \in N$. This is important for the algorithm for **(multi-quant-joint)** in Section E.

We define $f_N(a)$, lying between values $\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{P}^{\sigma}[A_t = a \mid \Omega_N]$ and $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{P}^{\sigma}[A_t = a \mid \Omega_N]$, 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 $a \in A$

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

whenever $\mathbb{P}^{\sigma}[\Omega_N] > 0$. In such a case, we set for all $a \in A$ and $N \subseteq [n]$,

$$x_{a,N} := f_N(a) \ ,$$

and $x_{a,N} := 0$ whenever $\mathbb{P}^{\sigma}[\Omega_N] = 0$. Since actions not in MECs are almost surely taken only finitely many times, we have

$$x_{a,N} = 0 \qquad \text{for } a \notin \bigcup \mathsf{MEC}, N \subseteq [n] \tag{1}$$

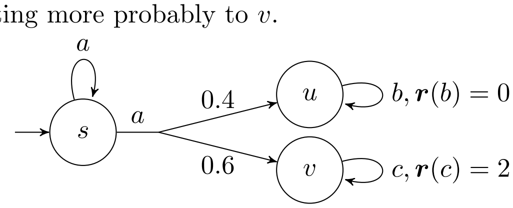
We show that (in)equations 4–7 of L are satisfied.

Equation 4 For $N \subseteq [n], t \in \mathbb{N}, a \in A, s \in S$, let

$$\Delta_t^N(a)(s) := \mathbb{P}^\sigma[S_{t+1} = s \mid A_t = a,\ \Omega_N]$$

denote the "transition probability" at time t restricted to runs in Ω_N. In general, $\Delta_i^N(a)(s)$ may be different from $\delta(a)(s)$. In such a case, the runs of Ω_N are "lucky" at the cost of someone else being "unlucky".

Example 9. Consider an action a with $\delta(a)(u) = 0.4$ and $\delta(a)(v) = 0.6$. It may well be that for instance $\Delta_1^\emptyset(a)(u) = 1$, but consequently, $\mathbb{P}^\sigma[\Omega_\emptyset] \leq 0.4$. This is exemplified in the figure below with $\boldsymbol{sat} = 1$. Intuitively, runs of $\Omega_\emptyset$ are extra-ordinarily lucky in getting to u, which must be compensated by other runs getting more probably to v.



More generally, if $\Delta_1^N(a)(u) = p \geq 0.4$ then $\mathbb{P}^\sigma[\Omega_N] \leq 0.4/p$. △

However, runs in Ω_N cannot be "lucky" all the time with positive probability:

Lemma 6. *Let $N \subseteq [n]$ be such that $\mathbb{P}^\sigma[\Omega_N] > 0$. Then for every $a \in A$ with $x_{a,N} > 0$, we have* $\lim\limits_{\substack{t\to\infty \\ \mathbb{P}^\sigma[A_t=a|\Omega_N]}} \Delta_t^N(a) = \delta(a)$.

Proof. Suppose for a contradiction, that the subsequence of $\Delta_t^N(a)(s)$, for steps t where a is positively used, does not converge to $\delta(a)(s)$. Then either $\limsup\limits_{\substack{t\to\infty \\ \mathbb{P}^\sigma[A_t=a|\Omega_N]}} = d > \delta(a)(s)$ or $\liminf\limits_{\substack{t\to\infty \\ \mathbb{P}^\sigma[A_t=a|\Omega_N]}} = d < \delta(a)(s)$. Without loss of generality, we assume the former. Then $\mathbb{P}^\sigma[A_t = a \text{ for infinitely many } t \mid \Omega_N] \leq \lim\limits_{t\to\infty} \left(\dfrac{\delta(a)(s)}{d}\right)^t = 0\,.$ Hence $f_N(a) = 0$, a contradiction with $x_{a,N} > 0$. □

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

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

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

$$\begin{aligned}
&\frac{1}{\mathbb{P}^\sigma[\Omega_N]} \sum_{a\in A} f_N(a) \cdot \delta(a)(s) \\
&= \frac{1}{\mathbb{P}^\sigma[\Omega_N]} \sum_{a\in A} \lim_{\ell\to\infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\sigma[A_t = a \ \mid \Omega_N] \cdot \mathbb{P}^\sigma[\Omega_N] \cdot \delta(a)(s)
\end{aligned}$$

(definition of f_N)

$$= \lim_{\ell\to\infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \sum_{a\in A} \mathbb{P}^\sigma[A_t = a \ \mid \varOmega_N] \cdot \delta(a)(s) \qquad \text{(linearity of the limit)}$$

$$= \lim_{\ell\to\infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \sum_{a\in A} \mathbb{P}^\sigma[A_t = a \ \mid \varOmega_N] \cdot \varDelta_t^N(a)(s) \quad \text{(Lemma 6 and Cesaro limit)}$$

$$= \lim_{\ell\to\infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\sigma[S_{t+1} = s \ \mid \varOmega_N] \qquad \text{(definition of } \delta)$$

$$= \lim_{\ell\to\infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\sigma[S_t = s \ \mid \varOmega_N] \qquad \text{(reindexing and Cesaro limit)}$$

$$= \lim_{\ell\to\infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \sum_{a\in Act(s)} \mathbb{P}^\sigma[A_t = a \ \mid \varOmega_N] \qquad (s \text{ must be followed by } a \in Act(s))$$

$$= \frac{1}{\mathbb{P}^\sigma[\varOmega_N]} \sum_{a\in Act(s)} \lim_{\ell\to\infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\sigma[A_t = a \ \mid \varOmega_N] \cdot \mathbb{P}^\sigma[\varOmega_N]$$

(linearity of the limit)

$$= \frac{1}{\mathbb{P}^\sigma[\varOmega_N]} \sum_{a\in Act(s)} f_N(a)\ . \qquad \text{(definition of } f_N)$$

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

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

where the second inequality is due to σ being a witness strategy and the first inequality follows from the following:

$$\sum_{N\subseteq[n]} \sum_{a\in A} x_{a,N} \cdot \boldsymbol{r}_i(a)$$

$$= \sum_{\substack{N\subseteq[n] \\ \mathbb{P}^\sigma[\varOmega_N]>0}} \sum_{a\in A} f_N(a) \cdot \boldsymbol{r}_i(a) \qquad \text{(definition of } x_{a,N})$$

$$= \sum_{\substack{N\subseteq[n] \\ \mathbb{P}^\sigma[\varOmega_N]>0}} \sum_{a\in A} \boldsymbol{r}_i(a) \cdot \lim_{\ell\to\infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\sigma[A_t = a \mid \varOmega_N] \cdot \mathbb{P}^\sigma[\varOmega_N] \quad \text{(definition of } f_N)$$

$$= \sum_{\substack{N\subseteq[n] \\ \mathbb{P}^\sigma[\varOmega_N]>0}} \mathbb{P}^\sigma[\varOmega_N] \cdot \lim_{\ell\to\infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \sum_{a\in A} \boldsymbol{r}_i(a) \cdot \mathbb{P}^\sigma[A_t = a \mid \varOmega_N]$$

(linearity of the limit)

$$\geq \sum_{\substack{N\subseteq[n]\\ \mathbb{P}^\sigma[\Omega_N]>0}} \mathbb{P}^\sigma[\Omega_N] \cdot \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{a\in A} \boldsymbol{r}_i(a) \cdot \mathbb{P}^\sigma[A_t = a \mid \Omega_N]$$

(definition of lim inf)

$$= \sum_{\substack{N\subseteq[n]\\ \mathbb{P}^\sigma[\Omega_N]>0}} \mathbb{P}^\sigma[\Omega_N] \cdot \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}^\sigma[\boldsymbol{r}_i(A_t) \mid \Omega_N]$$

(definition of the expectation)

$$\geq \sum_{\substack{N\subseteq[n]\\ \mathbb{P}^\sigma[\Omega_N]>0}} \mathbb{P}^\sigma[\Omega_N] \cdot \mathbb{E}^\sigma[\mathrm{lr}_{\inf}(\boldsymbol{r}_i) \mid \Omega_N] \qquad \text{(Fatou's lemma)}$$

$$= \mathbb{E}^\sigma[\mathrm{lr}_{\inf}(\boldsymbol{r}_i)] \qquad (\Omega_N\text{'s partition } \mathsf{Runs})$$

Although Fatou's lemma (see, e.g. [26, Chapter 4, Section 3]) requires the function $\boldsymbol{r}_i(A_t)$ be non-negative, we can replace it with the non-negative function $\boldsymbol{r}_i(A_t) - \min_{a\in A} \boldsymbol{r}_i(a)$ and add the subtracted constant afterwards.

Equation 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 trivially for $\mathbb{P}^\sigma[\Omega_N] = 0$, and whenever $\mathbb{P}^\sigma[\Omega_N] > 0$ we have

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

$$\geq \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{a\in C} \boldsymbol{r}_i(a) \cdot \mathbb{P}^\sigma[A_t = a \mid \Omega_N] \cdot \mathbb{P}^\sigma[\Omega_N]$$

(as above by def. of $x_{a,N}, f_N$, linearity of lim, def. of lim inf)

$$= \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{a\in C} \Big(\boldsymbol{r}_i(a) \cdot \mathbb{P}^\sigma[A_t = a \mid \Omega_N \cap \Omega_C] \cdot \frac{\mathbb{P}^\sigma[\Omega_N \cap \Omega_C]}{\mathbb{P}^\sigma[\Omega_N]} + \boldsymbol{r}_i(a) \cdot \mathbb{P}^\sigma[A_t = a \mid \Omega_N \setminus \Omega_C] \cdot \frac{\mathbb{P}^\sigma[\Omega_N \setminus \Omega_C]}{\mathbb{P}^\sigma[\Omega_N]} \Big) \cdot \mathbb{P}^\sigma[\Omega_N]$$

(partitioning of Runs)

$$= \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{a\in C} \boldsymbol{r}_i(a) \cdot \mathbb{P}^\sigma[A_t = a \mid \Omega_N \cap \Omega_C] \cdot \mathbb{P}^\sigma[\Omega_N \cap \Omega_C]$$

$$\Big(\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{P}^\sigma[A_t = a \mid \Omega_N \setminus \Omega_C] = 0 \text{ for } a \in C\Big)$$

$$\geq \mathbb{P}^\sigma[\Omega_N \cap \Omega_C] \cdot \mathbb{E}^\sigma[\mathrm{lr}_{\inf}(\boldsymbol{r}_i) \mid \Omega_N \cap \Omega_C]$$

(as above by def. of expectation and Fatou's lemma)

$$\geq \mathbb{P}^\sigma[\Omega_N \cap \Omega_C] \cdot \boldsymbol{sat}_i \qquad \text{(by definition of } \Omega_N \text{ and } i \in N\text{)}$$

It remains to prove the following:

Claim 1. *For $N \subseteq [n]$ and $C \in \mathsf{MEC}$, we have*

$$\sum_{a \in C} x_{a,N} = \mathbb{P}^\sigma[\Omega_N \cap \Omega_C] \; .$$

Proof.

$$\begin{aligned}
&\sum_{a \in C} x_{a,N} \\
&= \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in C} \mathbb{P}^\sigma[A_t = a \mid \Omega_N \cap \Omega_C] \cdot \mathbb{P}^\sigma[\Omega_N \cap \Omega_C] \\
&\hfill \text{(as above with factor } \boldsymbol{r}_i(a) \text{ left out)} \\
&= \mathbb{P}^\sigma[\Omega_N \cap \Omega_C] \cdot \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in C} \mathbb{P}^\sigma[A_t = a \mid \Omega_N \cap \Omega_C] \\
&\hfill \text{(linearity of the limit)} \\
&= \mathbb{P}^\sigma[\Omega_N \cap \Omega_C] \cdot \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{P}^\sigma[A_t \in C \mid \Omega_N \cap \Omega_C] \\
&\hfill \text{(taking two different action at time } t \text{ are disjoint events)} \\
&= \mathbb{P}^\sigma[\Omega_N \cap \Omega_C] \qquad (A_t \in C \text{ for all but finitely many } t \text{ on } \Omega_C)
\end{aligned}$$

$\square$

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

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

and the first term actually equals

$$\begin{aligned}
\sum_{N \subseteq [n] : i \in N} \sum_{a \in A} x_{a,N} &= \sum_{N \subseteq [n] : i \in N} \sum_{C \in \mathsf{MEC}} \sum_{a \in C} x_{a,N} = && \text{(by (1))} \\
&= \sum_{N \subseteq [n] : i \in N} \sum_{C \in \mathsf{MEC}} \mathbb{P}^\sigma[\Omega_N \cap \Omega_C] && \text{(by Claim 1)} \\
&= \sum_{N \subseteq [n] : i \in N} \mathbb{P}^\sigma[\Omega_N] && (\Omega_C\text{'s partition almost all } \mathsf{Runs})
\end{aligned}$$

C.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 [24, Theorem 9.3.8], which requires the machinery of deviation matrices. Instead, we can first simplify the behaviour of σ in the transient part to memoryless using [3] and then obtain y_χ directly, like in [14], 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 [3, 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 [3] is the same but with only a single value used for N. We denote the copy of each state s of G in $\bar{G}$ by $\bar{s}$.

Claim 2. *There is a strategy $\bar{\sigma}$ in $\bar{G}$ such that for every $C \in \mathsf{MEC}$ and $N \subseteq [n]$,*

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

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 [3, Lemma 4.6], we obtain a strategy σ' in G' such that $\sum_{s \in C} \mathbb{P}^{\sigma'}_{\hat{s}'}[\Diamond f_s] = \mathbb{P}^{\sigma}_{\hat{s}}[\Omega_C]$. We modify σ' into $\bar{\sigma}$. It behaves as σ', but instead of taking action a_s with probability p, we take each action $a_{s,N}$ with probability $p \cdot \frac{\mathbb{P}^{\sigma}_{\hat{s}}[\Omega_C \cap \Omega_N]}{\mathbb{P}^{\sigma}_{\hat{s}}[\Omega_C]}$. (For $\mathbb{P}^{\sigma}_{\hat{s}}[\Omega_C] = 0$, it is defined arbitrarily.) Then

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

$\square$

By [14, Theorem 3.2], there is a memoryless strategy $\bar{\sigma}$ satisfying the claim above such that

$$y_a := \sum_{t=1}^{\infty} \mathbb{P}^{\bar{\sigma}}_{\bar{s}}[A_t = a] \qquad \text{(for actions } a \text{ preserved in } \bar{G}\text{)}$$
$$y_{s,N} := \mathbb{P}^{\bar{\sigma}}_{\bar{s}_0}[\Diamond f_{s,N}]$$

are finite values satisfying Equations 1 and 2, and, moreover,

$$y_{s,N} \geq \sum_{s \in C} \mathbb{P}^{\bar{\sigma}}[\Diamond f_{s,N}] \,.$$

By Claim 2 for each $C \in \mathsf{MEC}$ we thus have

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

and summing up over all C and N we have

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

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

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

where the last equality follows by Claim 1, yielding Equation 3.

D Proof of Theorem 1, item 3: Solution to L induces witness strategy

Let $\bar{x}_{a,N}, \bar{y}_a, \bar{y}_{s,N}, s \in S, a \in A, N \subseteq [n]$ be a solution to the system L. We show how it effectively induces a witness strategy σ.

D.1 x-values and recurrent behaviour

To begin with, we show that x-values describe the recurrent behaviour only:

Lemma 7. *Let $\bar{x}_{a,N}, a \in A, N \subseteq [n]$ be a non-negative solution to Equation 4 of system L. Then for any fixed N, $X_N := \{s, a \mid \bar{x}_{a,N} > 0, a \in Act(s)\}$ is a union of end components.*

In particular, $X_N \subseteq \bigcup \mathsf{MEC}$, and for every $a \in A \setminus \bigcup \mathsf{MEC}$ and $N \subseteq [n]$, we have $\bar{x}_{a,N} = 0$.

Proof. Denoting $\bar{x}_{s,N} := \sum_{a \in Act(s)} \bar{x}_{a,N} = \sum_{a \in A} \bar{x}_a \cdot \delta(a)(s)$ for each $s \in S$, we can write

$$X_N = \{a \mid x_{a,N} > 0\} \cup \{s \mid \bar{x}_{s,N} > 0\}.$$

Firstly, we need to show that for all $a \in X_N$, whenever $\delta(a)(s') > 0$ then $s' \in X_N$. Since $\bar{x}_{s',N} \geq \bar{x}_{a,N} \cdot \delta(a)(s') > 0$, we have $s' \in X_N$.

Secondly, let there be a path from s to t in X_N. We need to show that there is a path from t to s in X_N. Assume the contrary and denote $T \subseteq X_N$ the set of states with no path to s in X_N; we assume $t \in T$. We write the path from s to t as $s \cdots s'bt' \cdots t$ where $s' \in X_N \setminus T$ and $t' \in T$. Then $b \in Act(s')$ and $\delta(b)(t') > 0$. By summing Equation 4 for all states in X_N we obtain firstly

$$\sum_{s \in X_N} \sum_{a \in A} \bar{x}_a \cdot \delta(a)(s) = \sum_{s \in X_N \setminus T} \sum_{a \in A} \bar{x}_a \cdot \delta(a)(s) = \sum_{s \in X_N \setminus T} \sum_{a \in Act(s)} \bar{x}_a$$

since no transition leads from T to $X_N \setminus T$; and secondly,

$$\sum_{s\in X_N}\sum_{a\in A}\bar{x}_a\cdot\delta(a)(s) = \sum_{s\in X_N}\sum_{a\in Act(s)}\bar{x}_a\cdot\delta(a)(s) \geq (\sum_{s\in X_N\setminus T}\sum_{a\in Act(s)}\bar{x}_a)+\bar{x}_{t',N} > \sum_{s\in X_N\setminus T}\sum_{a\in Act(s)}\bar{x}_a$$

again by summing Equation 4 and by

$$\sum_{a\in Act(t')}\bar{x}_{a,N} = \bar{x}_{t',N} = \sum_{a\in A}\bar{x}_{a,N}\cdot\delta(a,t') \geq \bar{x}_{b,N}\cdot\delta(b)(t') > 0$$

whence the contradiction. □

We thus start with the construction of the recurrent behaviour from x-values. For the moment, we restrict to strongly connected MDP and focus on Equation 4 for a particular fixed $N \subseteq [n]$. Note that for a fixed $N \subseteq [n]$ we have a system of equations equivalent to the form

$$\sum_{a\in A} x_a\cdot\delta(a)(s) = \sum_{a\in Act(s)} x_a \qquad \text{for each } s\in S. \tag{2}$$

We set out to prove Corollary 1. This crucial observation states 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.

Firstly, we construct a strategy for each "strongly connected" part of the solution $\bar{x}_a$ (each end-component of X_N of Lemma 7).

Lemma 8. *In a strongly connected MDP, let $\bar{x}_{a,N}, a\in A$ be a non-negative solution to Equation 4 of system L for a fixed $N\subseteq[n]$ and $\sum_{a\in A} x_{a,N} > 0$. It induces a memoryless strategy ζ such that for every BSCCs D of G^{ζ}, every $a\in D\cap A$, and almost all runs in D*

$$\boldsymbol{freq}_a = \bar{x}_{a,N}/\sum_{a\in D\cap A}\bar{x}_{a,N}$$

Proof. By [3, Lemma 4.3] applied on Equation (2), we get a memoryless strategy ζ such that $\mathbb{E}^{\zeta}[\boldsymbol{freq}_a \mid \Omega_D] = \bar{x}_{a,N}/\sum_{a\in D\cap A}\bar{x}_{a,N}$. Furthermore, by the ergodic theorem, $\boldsymbol{freq}_a$ returns the same vector for almost all runs in Ω_D, hence equal to $\mathbb{E}^{\zeta}[\boldsymbol{freq}_a]$. □

Secondly, we connect the parts (more end components of Lemma 7 within one MEC) and thus average the frequencies. This happens at a cost of small error used for transiting between the strongly connected parts.

Claim 3. *In a strongly connected MDP, let $\bar{x}_{a,N}, a\in A$ be a non-negative solution to Equation 4 of system L for a fixed $N\subseteq[n]$ and $\sum_{a\in A} x_{a,N} > 0$. For every $\varepsilon > 0$, there is a strategy ζ^{ε} such that for all $a\in A$ almost surely*

$$\boldsymbol{freq}_a > \frac{x_{a,N}}{\sum_{a\in A} x_{a,N}} - \varepsilon$$

Proof. We obtain ζ^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 [3, Proposition 5.1, Part 2].

There exists an arbitrarily small (strictly) positive solution x'_a of Equation (2). Indeed, it suffices to consider a strategy τ which always takes the uniform distribution over the actions in every state and then assign $\mathbb{E}^\tau[\boldsymbol{freq}_a/M]$ to x'_a for sufficiently large M. As the system of Equations (2) is linear and homogeneous, assigning $\bar{x}_{a,N}+x'_a$ to $x_{a,N}$ also solves this system (and thus Equation 4 as well) and Lemma 8 gives us a strategy ζ^ε satisfying almost surely

$$\boldsymbol{freq}_a = (\bar{x}_{a,N}+x'_a)/\sum_{a'\in A}\left(\bar{x}_{a',N}+x'_{a'}\right).$$

Moreover, since all actions are always taken with positive probability, we obtain a single BSCC and almost all runs thus have the same frequencies. We may safely assume that $\sum_{a\in A} x'_a \leq \frac{\varepsilon}{1-\varepsilon}\cdot\sum_{a\in A}\bar{x}_{a,N}$. Thus, we obtain

$$\boldsymbol{freq}_a > \bar{x}_{a,N}/\sum_{a\in A}\bar{x}_{a,N} - \varepsilon$$

almost surely (with $\mathbb{P}^{\zeta^\varepsilon}$-probability 1) by the following sequence of (in)equalities:

$$\begin{aligned}
&\boldsymbol{freq}_a \\
&= \frac{\bar{x}_{a,N}+x'_a}{\sum_{a\in A}(\bar{x}_{a,N}+x'_a)} && \text{(by Lemma 8)}\\
&> \frac{\bar{x}_{a,N}}{\sum_{a\in A}\bar{x}_{a,N}+\sum_{a\in A}x'_a} && \text{(by } x'_a>0)\\
&\geq \frac{\bar{x}_{a,N}}{\sum_{a\in A}\bar{x}_{a,N}+\frac{\varepsilon}{1-\varepsilon}\cdot\sum_{a\in A}\bar{x}_{a,N}} && \text{(by } \textstyle\sum_{a\in A}x'_a\leq\frac{\varepsilon}{1-\varepsilon}\cdot\sum_{a\in A}\bar{x}_{a,N})\\
&= \frac{\bar{x}_{a,N}}{\frac{1}{1-\varepsilon}\cdot\sum_{a\in A}\bar{x}_{a,N}} && \text{(rearranging)}\\
&= \frac{\bar{x}_{a,N}}{\sum_{a\in A}\bar{x}_{a,N}} - \varepsilon\cdot\frac{\bar{x}_{a,N}}{\sum_{a\in A}\bar{x}_{a,N}} && \text{(rearranging)}\\
&\geq \frac{\bar{x}_{a,N}}{\sum_{a\in A}\bar{x}_{a,N}} - \varepsilon && (\tfrac{\bar{x}_{a,N}}{\sum_{a\in A}\bar{x}_{a,N}}\leq 1)
\end{aligned}$$

□

Thirdly, we eliminate this error as we let the transiting (by x'_a) happen with probability vanishing over time.

Claim 4. *In a strongly connected MDP, let ξ_i be a sequence of strategies, each with $\boldsymbol{freq} = \boldsymbol{f}^i$ almost surely, and such that $\lim_{i\to\infty}\boldsymbol{f}^i$ is well defined. Then there is a strategy ξ such that almost surely*

$$\boldsymbol{freq} = \lim_{i\to\infty}\boldsymbol{f}^i\,.$$

Proof. This proof very closely follows the computation in [3, Proposition 5.1, Part "Moreover"], but for general ξ_i.

Given $a \in A$, let $\mathit{lf}_a := \lim_{i\to\infty} \boldsymbol{f}^i_a$. By definition of limit and the assumption that $\boldsymbol{freq}_a = \mathrm{lr}_{\inf}(\mathbb{1}_a)$ is almost surely equal to $\boldsymbol{f}^i_a$ for each ξ_i, there is a subsequence ξ_j of the sequence ξ_i such that $\mathbb{P}^{\xi_j}\left[\mathrm{lr}_{\inf}(\mathbb{1}_a) \geq \mathit{lf}_a - 2^{-j-1}\right] = 1$. 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

$$\mathbb{P}^{\xi_j}\left[\inf_{T \geq \kappa_j} \frac{1}{T} \sum_{t=0}^{T} \mathbb{1}_a(A_t) \geq \mathit{lf}_a - 2^{-j}\right] \geq 1 - 2^{-j}.$$

Let us consider a sequence $n_0, n_1, \dots$ of numbers where $n_j \geq \kappa_j$ and

$$\frac{\sum_{k<j} n_j}{n_j} \leq 2^{-j} \tag{3}$$

$$\frac{\kappa_{j+1}}{n_j} \leq 2^{-j} \tag{4}$$

We define ξ to behave as ξ_1 for the first n_1 steps, then as ξ_2 for the next n_2 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 $n_0, n_1, \dots$ 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. $\mathbb{1}_a$ is arbitrarily close to lf_a infinitely often. To get that lf_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 lf_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

$$\liminf_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \mathbb{1}_a(A_t(\omega)) \geq \mathit{lf}_a\,.$$

Denote by E_k the set of all runs $\omega = s_0 a_0 s_1 a_1 \cdots$ of G^ξ such that for some $\kappa_k \leq d \leq n_k$ we have

$$\frac{1}{d} \sum_{j=N_j}^{N_j+d} \mathbb{1}_a(a_k) \quad < \quad \mathit{lf}_a - 2^{-k}.$$

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 [26], almost surely only finitely many of E_j take place. Thus, almost every run $\omega = s_0 a_0 s_1 a_1 \cdots$ of G^ξ satisfies the following: there is ℓ such that for all $j \geq \ell$ and all $\kappa_j \leq d \leq n_j$ we have that

$$\frac{1}{d}\sum_{k=N_j}^{N_j+d} \mathbb{1}_a(a_k) \quad \geq \quad \mathit{lf}_a - 2^{-j}\,. \tag{5}$$

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

$$\frac{1}{T}\sum_{t=0}^{T} \mathbb{1}_a(a_t) \quad \geq \quad (\mathit{lf}_a - 2^{-j})(1 - 2^{1-j})\,. \tag{6}$$

Taking the limit of (6) where T (and thus also j) goes to ∞, we obtain

$$\boldsymbol{freq}_a(\omega) = \liminf_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T} \mathbb{1}_a(a_t) \geq \liminf_{j\to\infty} (\mathit{lf}_a - 2^{-j})(1-2^{1-j}) = \mathit{lf}_a = \lim_{i\to\infty} \boldsymbol{f}_a^i$$

yielding the claim. It remains to prove (6). First, note that

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

and that by (5)

$$\frac{1}{T}\sum_{t=N_{j-1}}^{N_j-1} \mathbb{1}_a(a_t) = \frac{1}{n_j}\sum_{t=N_{j-1}}^{N_j-1} \mathbb{1}_a(a_t) \cdot \frac{n_j}{T} \geq (\mathit{lf}_a - 2^{-j})\frac{n_j}{T}$$

which gives

$$\frac{1}{T}\sum_{t=0}^{T} \mathbb{1}_a(a_t) \;\geq\; (\mathit{lf}_a - 2^{-j})\frac{n_j}{T} \;+\; \frac{1}{T}\sum_{t=N_j}^{T} \mathbb{1}_a(a_t). \tag{7}$$

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

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

by (3) and (4). Therefore, by (7),

$$\frac{1}{T}\sum_{t=0}^{T} \mathbb{1}_a(a_t) \quad \geq \quad (\mathit{lf}_a - 2^{-j})(1 - 2^{1-j}).$$

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

$$\frac{1}{T}\sum_{t=N_j}^{T} \mathbb{1}_a(a_t) = \frac{1}{T - N_j}\sum_{t=N_j}^{T} \mathbb{1}_a(a_t) \cdot \frac{T - N_j}{T}$$

$$\geq (\mathit{lf}_a - 2^{-j}) \left(1 - \frac{N_{j-1} + n_j}{T}\right) \qquad \text{(by (5))}$$
$$\geq (\mathit{lf}_a - 2^{-j}) \left(1 - 2^{-j} - \frac{n_j}{T}\right) \qquad \text{(by (3))}$$

and thus, by (7),

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T} \mathbb{1}_a(a_t) &\geq (\mathit{lf}_a - 2^{-j})\frac{n_j}{T} + (\mathit{lf}_a - 2^{-j-1}) \left(1 - 2^{-j} - \frac{n_j}{T}\right) \\ &\geq (\mathit{lf}_a - 2^{-j}) \left(\frac{n_j}{T} + \left(1 - 2^{-j} - \frac{n_j}{T}\right)\right) \\ &\geq (\mathit{lf}_a - 2^{-j})(1 - 2^{-j})\end{aligned}$$

which finishes the proof of (6). □

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.

Corollary 1. *For a strongly connected MDP, let $\bar{x}_{a,N}, a \in A$ be a non-negative solution to Equation 4 of system L for a fixed $N \subseteq [n]$ and $\sum_{a \in A} x_{a,N} > 0$. Then there is a strategy ξ_N such that for each $a \in A$ almost surely*

$$\boldsymbol{freq}_a = \frac{x_{a,N}}{\sum_{a \in A} x_{a,N}} \,.$$

Proof. The strategy ξ_N is constructed by Claim 4 taking ξ_i to be $\zeta^{1/i}$ from Claim 3. □

Since the fraction is independent of the initial state of the MDP, the frequency is almost surely the same also for all initial states. The reward of ξ_N is almost surely

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

When the MDP is not strongly connected, we obtain such ξ_N in each MEC C with $\sum_{a \in C} \bar{x}_{a,N} > 0$ and the respective reward of almost all runs in C is thus

$$\mathbb{E}^{\xi_N}[\mathrm{lr}(\boldsymbol{r}) \mid \Omega_C] = \sum_{a \in C \cap A} \bar{x}_{a,N} \cdot \boldsymbol{r}(a) / \sum_{a \in C \cap A} \bar{x}_{a,N} \tag{8}$$

$$\mathbb{P}^{\xi_N}\left[\mathrm{lr}(\boldsymbol{r}) = \sum_{a \in C \cap A} \bar{x}_{a,N} \cdot \boldsymbol{r}(a) / \sum_{a \in C \cap A} \bar{x}_{a,N} \mid \Omega_C\right] = 1 \tag{9}$$

Finally, note that although ξ_N uses infinite memory, it only needs a counter and to know the current state.

D.2 y-values and transient behaviour

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

Lemma 9. *Let $\xi_N, N \subseteq [n]$ be strategies. Then every non-negative solution $\bar{y}_a, \bar{y}_{s,N}, a \in A, s \in S, N \subseteq [n]$ to Equation 1 effectively induces a strategy σ such that*

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

and σ is memoryless before the switch.

Proof. The idea is similar to [3, Proposition 4.2, Step 1]. However, instead of switching in s to ξ with some probability p, here we have to branch this decision and switch to ξ_N with probability $p \cdot \frac{\bar{y}_{s,N}}{\sum_{N\subseteq[n]} \bar{y}_{s,N}}$.

Formally, 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 [3] we have a stochastic-update strategy ϑ which stays eventually in each MEC C with probability y_C.

Then the strategy $\bar{\sigma}$ works as follows. It plays according to ϑ until a BSCC of G^ϑ is reached. This means that every possible continuation of the path stays in the current MEC C of G. Assume that C has states $s_1, \ldots, s_k$. 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 behaviour as follows: First, it 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, it chooses 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 on, 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 ϑ 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 $\bar{x}_{s_i,N}$. Further, as in [3] we can transform the part of $\bar{\sigma}$ before switching to ξ_N to a memoryless strategy and thus get strategy σ. □

Claim 5. *Let $\xi_N, N \subseteq [n]$ be strategies. Then every non-negative solution $\bar{y}_a, \bar{y}_{s,N}, \bar{x}_{a,N}, a \in A, s \in S, N \subseteq [n]$ to Equations 1 and 3 effectively induces a strategy σ such that for every MEC C*

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

and σ is memoryless before the switch.

Proof. By Lemma 9 and Equation 3. □

D.3 Proof of witnessing

We now prove that the strategy σ of Claim 5 with $\xi_N, N \subseteq [n]$ of Corollary 1 is indeed a witness strategy. Note that existence of xi_N's depends on the sums of $\bar{x}$-values being positive. This follows by Equation 2 and 3. We evaluate the strategy σ as follows:

$$
\begin{aligned}
\mathbb{E}^\sigma[\mathrm{lr}_{\inf}(\boldsymbol{r})] &= \sum_{C \in \mathsf{MEC}} \sum_{N \subseteq [n]} \mathbb{P}^\sigma[\text{switch to } \xi_N \text{ in } C] \cdot \mathbb{E}^{\xi_N}[\mathrm{lr}(\boldsymbol{r}) \mid \Omega_C] \\
&\qquad\qquad\qquad \text{(by Equation 2, } \sum_{N \subseteq [n]} \mathbb{P}^\sigma[\text{switch to } \xi_N] = 1) \\
&= \sum_{C \in \mathsf{MEC}} \sum_{N \subseteq [n]} \Big(\sum_{a \in C \cap A} \bar{x}_{a,N} \Big) \cdot \mathbb{E}^{\xi_N}[\mathrm{lr}(\boldsymbol{r}) \mid \Omega_C] \qquad \text{(by Claim 5)} \\
&= \sum_{C \in \mathsf{MEC}} \sum_{\substack{N \subseteq [n]: \\ \sum_{a \in C \cap A} \bar{x}_{a,N} > 0}} \Big(\sum_{a \in C \cap A} \bar{x}_{a,N} \Big) \cdot \Big(\sum_{a \in C \cap A} \bar{x}_{a,N} \cdot \boldsymbol{r}(a) / \sum_{a \in C \cap A} \bar{x}_{a,N} \Big) \\
&\qquad\qquad\qquad \text{(by (8))} \\
&= \sum_{N \subseteq [n]} \sum_{C \in \mathsf{MEC}} \sum_{a \in C \cap A} \bar{x}_{a,N} \cdot \boldsymbol{r}(a) \\
&= \sum_{N \subseteq [n]} \sum_{a \in A \cap \bigcup \mathsf{MEC}} \bar{x}_{a,N} \cdot \boldsymbol{r}(a) \\
&= \sum_{N \subseteq [n]} \sum_{a \in A} \bar{x}_{a,N} \cdot \boldsymbol{r}(a) \qquad \text{(by Lemma 7)} \\
&\geq \boldsymbol{exp} \qquad \text{(by Equation 5)}
\end{aligned}
$$

and for each $i \in [n]$

$$
\begin{aligned}
\mathbb{P}^\sigma[\mathrm{lr}(\boldsymbol{r})_i \geq \boldsymbol{sat}_i] &= \sum_{C \in \mathsf{MEC}} \sum_{N \subseteq [n]} \mathbb{P}^\sigma[\text{switch to } \xi_N \text{ in } C] \cdot \mathbb{P}^{\xi_N}[\mathrm{lr}(\boldsymbol{r})_i \geq \boldsymbol{sat}_i \mid \Omega_C] \\
&\qquad\qquad\qquad \text{(by Equation 2, } \sum_{N \subseteq [n]} \mathbb{P}^\sigma[\text{switch to } \xi_N] = 1) \\
&= \sum_{C \in \mathsf{MEC}} \sum_{N \subseteq [n]} \Big(\sum_{a \in C \cap A} \bar{x}_{a,N} \Big) \cdot \mathbb{P}^{\xi_N}[\mathrm{lr}(\boldsymbol{r})_i \geq \boldsymbol{sat}_i \mid \Omega_C] \\
&\qquad\qquad\qquad \text{(by Claim 5)} \\
&= \sum_{C \in \mathsf{MEC}} \sum_{\substack{N \subseteq [n]: \\ \sum_{a \in C \cap A} \bar{x}_{a,N} > 0}} \Big(\sum_{a \in C \cap A} \bar{x}_{a,N} \Big) \cdot \mathbb{P}^{\xi_N} \Big[\sum_{a \in C \cap A} \bar{x}_{a,N} \cdot \boldsymbol{r}(a) / \sum_{a \in C \cap A} \bar{x}_{a,N} \geq \boldsymbol{sat}_i \Big] \\
&\qquad\qquad\qquad \text{(by (9))}
\end{aligned}
$$

$$\geq \sum_{C \in \mathsf{MEC}} \sum_{\substack{i \in N \subseteq [n]: \\ \sum_{a \in C \cap A} \bar{x}_{a,N} > 0}} \Big(\sum_{a \in C \cap A} \bar{x}_{a,N} \Big) \cdot \mathbb{P}^{\xi_N} \left[\sum_{a \in C \cap A} \bar{x}_{a,N} \cdot \boldsymbol{sat}_i / \sum_{a \in C \cap A} \bar{x}_{a,N} \geq \boldsymbol{sat}_i \right]$$

(by Equation 6)

$$= \sum_{i \in N \subseteq [n]} \sum_{C \in \mathsf{MEC}} \sum_{a \in C \cap A} \bar{x}_{a,N}$$

$$= \sum_{i \in N \subseteq [n]} \sum_{a \in A \cap \bigcup \mathsf{MEC}} \bar{x}_{a,N}$$

$$= \sum_{i \in N \subseteq [n]} \sum_{a \in A} \bar{x}_{a,N} \qquad \text{(by Lemma 7)}$$

$$\geq \boldsymbol{pr}_i \qquad \text{(by Equation 7)}$$

E Complexity issues

In this section, we comment in more detail on claims of Remark 1, Section 5.1, and Section 5.3.

E.1 Linear program for (multi-quant-joint)

The program for **(multi-quant-joint)** arises from the one for **(multi-quant-conjunctive)** of Fig. 4 by considering only the values $\emptyset$ and $[n]$ for the index N. Indeed, $[n]$-good runs are exactly the ones satisfying jointly the satisfaction value thresholds.

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 $s \in S$

$$\mathbf{1}_{s_0}(s) + \sum_{a \in A} y_a \cdot \delta(a)(s) = \sum_{a \in Act(s)} y_a + y_{s,\emptyset} + y_{s,[n]}$$

2. almost-sure switching to recurrent behaviour:

$$\sum_{s \in C} y_{s,\emptyset} + y_{s,[n]} = 1$$

3. probability of switching in a MEC is the frequency of using its actions: for $C \in \mathsf{MEC}$

$$\sum_{s \in C} y_{s,\emptyset} = \sum_{a \in C} x_{a,\emptyset}$$

$$\sum_{s \in C} y_{s,[n]} = \sum_{a \in C} x_{a,[n]}$$

4. recurrent flow: for $s \in S$

$$\sum_{a \in A} x_{a,\emptyset} \cdot \delta(a)(s) = \sum_{a \in Act(s)} x_{a,\emptyset}$$

$$\sum_{a \in A} x_{a,[n]} \cdot \delta(a)(s) = \sum_{a \in Act(s)} x_{a,[n]}$$

5. expected rewards:

$$\sum_{a \in A} \Big(x_{a,\emptyset} + x_{a,[n]} \Big) \cdot \boldsymbol{r}(a) \geq \boldsymbol{exp}$$

6. commitment to satisfaction: for $C \in \mathsf{MEC}$ and $i \in [n]$

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

7. satisfaction: for $i \in [n]$

$$\sum_{a \in A} x_{a,[n]} \geq pr$$

Similarly, for **(mono-quant)** it is sufficient to consider $N = [n] = \{1\}$ and $N = \emptyset$ only. Consequently, for **(multi-qual)** $N = [n]$, and for **(mono-qual)** $N = [n] = \{1\}$ are sufficient, thus the index N can be removed completely.

E.2 Linear program for the combined problem with (EXP), (conjunctive-SAT), and (joint-SAT) constraints

Here we consider the problem **(multi-quant-conjunctive)** as defined in Section 2.2 with an additional (joint-SAT) constraint

$$\mathbb{P}^{\sigma}\Big[\mathrm{lr}_{\mathrm{inf}}(\boldsymbol{r}) \geq \widetilde{\boldsymbol{sat}}\Big] \geq \widetilde{pr}$$

for $\widetilde{pr} \in [0,1] \cap \mathbb{Q}$ and $\widetilde{\boldsymbol{sat}} \in \mathbb{Q}^n$. The linear program for this "combined" problem can be easily derived from the program L in Fig. 4 as follows.

The first step consists in splitting the recurrent flow into two parts, *yes* and *no* (requiring all variables be non-negative):

1. transient flow: for $s \in S$

$$\mathbf{1}_{s_0}(s) + \sum_{a \in A} y_a \cdot \delta(a)(s) = \sum_{a \in Act(s)} y_a + \sum_{N \subseteq [n]} (y_{s,N,yes} + y_{s,N,no})$$

2. almost-sure switching to recurrent behaviour:

$$\sum_{\substack{s \in C \in \mathsf{MEC} \\ N \subseteq [n]}} (y_{s,N,yes} + y_{s,N,no}) = 1$$

3. probability of switching in a MEC is the frequency of using its actions: for $C \in \mathsf{MEC}, N \subseteq [n]$

$$\sum_{s \in C} y_{s,N,yes} = \sum_{a \in C} x_{a,N,yes}$$

$$\sum_{s \in C} y_{s,N,no} = \sum_{a \in C} x_{a,N,no}$$

4. recurrent flow: for $s \in S, N \subseteq [n]$

$$\sum_{a \in A} x_{a,N,yes} \cdot \delta(a)(s) = \sum_{a \in Act(s)} x_{a,N,yes}$$

$$\sum_{a \in A} x_{a,N,no} \cdot \delta(a)(s) = \sum_{a \in Act(s)} x_{a,N,no}$$

5. expected rewards:

$$\sum_{\substack{a \in A,\\ N \subseteq [n]}} (x_{a,N,yes} +_{a,N,no}) \cdot \boldsymbol{r}(a) \geq \boldsymbol{exp}$$

6. commitment to satisfaction: for $C \in \mathsf{MEC}$, $N \subseteq [n]$, $i \in N$

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

$$\sum_{a \in C} x_{a,N,no} \cdot \boldsymbol{r}(a)_i \geq \sum_{a \in C} x_{a,N,no} \cdot \boldsymbol{sat}_i$$

7. satisfaction: for $i \in [n]$

$$\sum_{\substack{a \in A,\\ N \subseteq [n]: i \in N}} x_{a,N,yes} \geq \boldsymbol{pr}_i$$

$$\sum_{\substack{a \in A,\\ N \subseteq [n]: i \in N}} x_{a,N,no} \geq \boldsymbol{pr}_i$$

Note that this program has the same set of solutions as the original program, considering substitution $\alpha_{\beta,N} = \alpha_{\beta,N,yes} + \alpha_{\beta,N,no}$.

The second step consists in using the "*yes*" part of the flow for ensuring satisfaction of the (joint-SAT) constraint. Formally, we add the following additional equations (of type 6 and 7):

$$\sum_{a \in C} x_{a,N,yes} \cdot \boldsymbol{r}(a)_i \geq \sum_{a \in C} x_{a,N,yes} \cdot \widetilde{\boldsymbol{sat}}_i \qquad \text{for } i \in [n] \text{ and } N \subseteq [n]$$

$$\sum_{\substack{a \in A\\ N \subseteq [n]}} x_{a,N,yes} \geq \widetilde{pr}$$

Since the number of variables is double that for **(multi-quant-conjunctive)**, the complexity remains essentially the same: polynomial in the size of the MDP and exponential in n.

Furthermore, we can also allow multiple constraints, i.e. more (joint-SAT) constraints or more (conjunctive-SAT), thus specifying probability thresholds for more value thresholds for each reward. Then instead of subsets of $[n]$ as so far, we consider subsets of the set of all constraints. The complexity is then exponential in the number of constraints rather than just in the dimension of the rewards.

E.3 Optimization of satisfaction

Similarly to the proof of Theorem 5, we can obtain a Pareto-curve approximation for possible values of the $\boldsymbol{sat}$ or $\boldsymbol{pr}$ vectors for a given $\boldsymbol{exp}$ vector. We simply replace these vectors by vectors of variables, obtaining a multi-objective linear program. If we want the complete Pareto-curve approximation for all the parameters $\boldsymbol{sat}$, $\boldsymbol{pr}$, and $\boldsymbol{exp}$, the number of objectives rises from n to $3 \cdot n$. The complexity is thus still polynomial in the size of the MDP and $1/\varepsilon$, and exponential in n.

In particular, for the single-reward case, we can compute also the optimal $\boldsymbol{pr}$ given $\boldsymbol{exp}$ and $\boldsymbol{sat}$, or the optimal $\boldsymbol{sat}$ given $\boldsymbol{pr}$ and $\boldsymbol{exp}$.

E.4 Proof of Theorem 3

Theorem 3. *The realizability problem of conjunction of (joint-SAT) and (conjunctive-SAT) is NP-hard (even without the (EXP) constraint).*

Proof. We proceed by reduction from SAT. Let φ be a formula with the set of clauses $C = \{c_1, \dots, c_k\}$ over atomic propositions $Ap = \{a_1, \dots, a_p\}$. We denote $\overline{Ap} = \{\bar{a}_1, \dots, \bar{a}_p\}$ the literals that are negations of the atomic propositions.

We define an MDP $G_\varphi = (S, A, Act, \delta, \hat{s})$ as follows:

- $S = \{s_i \mid i \in [p]\}$,
- $A = Ap \cup \overline{Ap}$,
- $Act(s_i) = \{a_i, \bar{a}_i\}$ for $i \in [p]$,
- $\delta(a_i)(s_{i+1})$ and $\delta(\bar{a}_i)(s_{i+1}) = 1$ (actions are assigned Dirac distributions),
- $\hat{s} = s_1$.

Intuitively, a run in G_φ repetitively chooses a valuation.

We define the dimension of the reward function to be $n = k + 2p$. We index the components of vectors with this dimension by $C \cup Ap \cup \overline{Ap}$. The reward function is defined for each $\ell \in A$ as follows:

- $\boldsymbol{r}(\ell)(c_i) = \begin{cases} 1 & \text{if } \ell \models c_i \\ 0 & \text{if } \ell \not\models c_i \end{cases}$

– $\boldsymbol{r}(\ell)(a_i) = \begin{cases} 1 & \text{if } \ell = a_i \\ -1 & \text{if } \ell = \bar{a}_i \\ 0 & \text{otherwise} \end{cases}$

– $\boldsymbol{r}(\ell)(\bar{a}_i) = \begin{cases} -1 & \text{if } \ell = a_i \\ 1 & \text{if } \ell = \bar{a}_i \\ 0 & \text{otherwise} \end{cases}$

Intuitively, we get a positive reward for a clause when it is guaranteed to be satisfied by the choice of a literal. The latter two items count number of uses of a literal; the count is reduced by every occurrence of the opposite literal.

The realizability problem instance R_φ is then defined by a conjunction of the following (conjunctive-SAT) and (joint-SAT) constraints:

$$\mathbb{P}^\sigma \left[\mathrm{lr}_{\mathrm{inf}}(\boldsymbol{r})_\ell \geq \frac{1}{k} \right] \geq \frac{1}{2} \qquad \text{for each } \ell \in Ap \cup \overline{Ap} \tag{conjunctive-S}$$

$$\mathbb{P}^\sigma \left[\bigwedge_{c \in C} \mathrm{lr}_{\mathrm{inf}}(\boldsymbol{r})_c \geq 1 \right] \geq \frac{1}{2} \tag{joint-S}$$

Intuitively, (conjunctive-S) ensures that almost all runs choose, for each atomic proposition, either the positive literal with frequency 1, or the negative literal with frequency 1; in other words, it ensures that the choice of valuation is consistent within the run almost surely. Indeed, since the choice between a_i and $\bar{a}_i$ happens every k steps, runs that mix both with positive frequency cannot exceed the value threshold $1/k$. Therefore, half of the runs must use only a_i, half must use only $\bar{a}_i$. Consequently, almost all runs choose one of them consistently.

Further, (joint-S) on the top ensures that there is a (consistent) valuation that satisfies all the clauses. Moreover, we require that this valuation is generated with probability at least 1/2. Actually, we only need probability strictly greater than 0.

We now prove that φ is satisfiable if and only if the problem instance defined above on MDP G_φ is realizable.

"Only if part": Let $\nu \subseteq Ap \cup \overline{Ap}$ be a satisfying valuation for φ. We define σ to have initial distribution on memory elements m_1, m_2 with probability 1/2 each. With memory m_1 we always choose action from ν and with memory m_2 from the "opposite valuation" $\overline{\nu}$ (where $\bar{\bar{a}}$ is identified with a).

Therefore, each literal has frequency $1/k$ either in the first or the second kind of runs. Further, the runs of the first kind (with memory m_1) satisfy all clauses.

"If part": Given a witness strategy σ for $R(\varphi)$, we construct a satisfying valuation. First, we focus on the property induced by the (conjunctive-S) constraint. We show that almost all runs uniquely induce a valuation

$$\nu_\sigma := \{\ell \in Ap \cup \overline{Ap} \mid \mathrm{freq}_\ell > 0\}$$

which follows from the following lemma:

Lemma 10. *For every witness strategy σ satisfying the (conjunctive-S) constraint, and for each $a \in Ap$, we have*

$$\mathbb{P}^{\sigma}\left[\mathrm{freq}_a = \frac{1}{k} \text{ and } \mathrm{freq}_{\bar{a}} = 0\right] + \mathbb{P}^{\sigma}\left[\mathrm{freq}_a = 0 \text{ and } \mathrm{freq}_{\bar{a}} = \frac{1}{k}\right] = 1\,.$$

Proof. Let $a \in Ap$ be an arbitrary atomic proposition. To begin with, observe that due to the circular shape of MDP G_{φ}, we have

$$\mathrm{freq}_a + \mathrm{freq}_{\bar{a}} = 1/k \tag{10}$$

for every run. Therefore, the two events $\mathrm{freq}_a \geq 1/k$ and $\mathrm{freq}_{\bar{a}} \geq 1/k$ are disjoint. Consequently, the two events $\mathrm{lr}_{\mathrm{inf}}(\boldsymbol{r})_a \geq 1/k$ and $\mathrm{lr}_{\mathrm{inf}}(\boldsymbol{r})_{\bar{a}} \geq 1/k$ are disjoint. Due to the (conjunctive-S) constraint, almost surely exactly one of the events occurs. By (10), almost surely either $\mathrm{freq}_a = \frac{1}{k}$ and $\mathrm{freq}_{\bar{a}} = 0$, or $\mathrm{freq}_a = 0$ and $\mathrm{freq}_{\bar{a}} = \frac{1}{k}$. □

By the (joint-S) constraint, we have a set Ω_{sat}, with non-zero measure, of runs satisfying $\mathrm{lr}_{\mathrm{inf}}(\boldsymbol{r})_c \geq 1$ for each $c \in C$. By the previous lemma, almost all runs of Ω_{sat} induce unique valuations. Since there are finitely many valuation, at least one of them is induced by a set of non-zero measure. Let ω be one of the runs and ν the corresponding valuation. We claim that ν is a satisfying valuation for φ.

Let $c \in C$ be any clause, we show $\nu \models c$. Since $\mathrm{lr}_{\mathrm{inf}}(\boldsymbol{r})(\omega)_c \geq 1$, there is an action ℓ such that

- $\mathrm{freq}_{\ell}(\omega) > 0$, and
- $\boldsymbol{r}(a)_{\ell} \geq 1$.

The former inequality implies that $\ell \in \nu$ and the latter that $\ell \models c$. Altogether, $\nu \models c$ for every $c \in C$, hence ν witnesses satisfiability of φ. □