Logical Methods in Computer Science
Vol. 13(2:15)2017, pp. 1–50
www.lmcs-online.org

Submitted Aug. 21, 2015
Published Jun. 30, 2017

UNIFYING TWO VIEWS ON MULTIPLE MEAN-PAYOFF OBJECTIVES IN MARKOV DECISION PROCESSES *

KRISHNENDU CHATTERJEE [a], ZUZANA KŘETÍNSKÁ [b], AND JAN KŘETÍNSKÝ [c]

[a] IST Austria
e-mail address: Krishnendu.Chatterjee@ist.ac.at

[b,c] Institut für Informatik, Technische Universität Muñchen, Germany
e-mail address: komarkova.zuza@gmail.com, jan.kretinsky@gmail.com

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) [How60, Put94, FV97]. An MDP consists of a finite state space, and in every state a controller can choose among several actions (the nondeterministic choices), and given the current state and the chosen action the system evolves stochastically according to a probabilistic transition function. Every 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

* This is an extended version of the LICS'15 paper with full proofs and additional complexity results.

This research was funded in part by Austrian Science Fund Grant No P 23499-N23, European Research Council Grant No 279307 (Graph Games), the DFG Research Training Group PUMA: Programm- und Modell-Analyse (GRK 1480), the Czech Science Foundation grant No. 15-17564S, and People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) REA Grant No 291734.

DOI:10.23638/LMCS-13(2:15)2017

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

Two views. There exist two views in the study of MDPs with mean-payoff objectives [BBC$^+$14]. The traditional and classical view is the *expectation* semantics, where the goal is to maximize (or minimize) the expectation of the mean-payoff objective. There are numerous applications of MDPs with the expectation semantics, such as in inventory control, planning, and performance evaluation [Put94, FV97]. The alternative semantics is called the *satisfaction* semantics, which, given a mean-payoff value threshold *sat* and a probability threshold *pr*, asks for a strategy to ensure that the 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 [BBC$^+$14]. While the joint interpretation of satisfaction has already been introduced and studied in [BBC$^+$14], here we consider also the conjunctive interpretation, which was not considered in [BBC$^+$14]. The conjunctive interpretation was considered in [FKR95], however, only a partial solution was provided, and it was mentioned that a complete solution would be very useful.

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:* a single reward function and qualitative satisfaction semantics;
(2) *Mono-quant:* a single reward function and quantitative satisfaction semantics;
(3) *Multi-qual:* multiple reward functions and qualitative satisfaction semantics;
(4) *Multi-quant:* 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, separately (*multi-quant-conjunctive*, *multi-quant-joint*) as well as at once (*multi-quant-conjunctive-joint*).

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 [BFRR14], whereas we consider probabilistic (both qualitative and quantitative) guarantee for risk aversion. We now illustrate our problem with several examples.

Illustrative examples:

- For simple risk aversion, consider a single reward function modelling investment. Positive reward stands for profit, negative for loss. We aim at maximizing the expected long-run average while guaranteeing that it is 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 [Put94, 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 exceed 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. For the gratis service, expected throughput $1Mbps$ is guaranteed as well as 60% connections running on at least $0.8Mbps$. 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 $5Mbps$ and 80% on even $8Mbps$ (satisfaction constraints). 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. Note that since the percentages above are different, the constraints cannot be encoded using the joint interpretation, and conjunctive interpretation is necessary.

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 component-wise ordering) that are not strictly dominated by any achievable solution. Pareto optimality has been studied in cooperative game theory [Owe95] and in multi-criterion optimization and decision making in both economics and engineering [Kos88, YC03, SCK04].

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

- *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[(*)] strategies are sufficient [FV97, BBE10, BBC$^+$14]. We show this carries over in the *mono-qual* case only. In contrast, we show that for *mono-quant* both randomization and memory is necessary. For randomized strategies, they can be *stochastic-update*, where the memory is updated probabilistically, or *deterministic-update*, where the memory update is deterministic. We provide precise bounds on the memory size of stochastic-update strategies. Further, we show that for both mono-quant and multi-qual, deterministic-update strategies require memory size that is dependent on the MDP. Finally, we also show that deterministic-update strategies are sufficient even for *multi-quant*, thus extending the results of [BBC$^+$14].
- *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 $\varepsilon > 0$.

[(*)]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.

In summary, we unify the two existing semantics, present comprehensive results related to algorithmic and strategy complexities for the unifying semantics, and 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 [CMH06, EKVY08, FKN$^+$11, BBC$^+$14] we also obtain our results by showing that the set of achievable solutions can be represented by a set of linear constraints, and from the linear constraints witness strategies for achievable solutions can be constructed. However, previous work on the satisfaction semantics [BBC$^+$14, RRS15] reduces the problem to invoking linear-programming solution for each maximal end-component and a separate linear program to combine the partial results together. In contrast, 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. 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* [Put94, Alt99]. 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 [Cha07] and a complete picture was presented in [BBC$^+$14]. The expectation and satisfaction semantics was considered in [BBC$^+$14], and our work unifies the two different semantics for MDPs. For general MDPs, [CMH06, CFW13] studied multiple discounted reward functions. MDPs with multiple ω-regular specifications were studied in [EKVY08]. It was shown that the Pareto curve can be approximated in polynomial time in the size of MDP and exponential in the number of specifications; the algorithm reduces the problem to MDPs with multiple reachability specifications, which can be solved by multi-objective linear programming [PY00]. In [FKN$^+$11], the results of [EKVY08] were extended to combine ω-regular and expected total reward objectives. The problem of conjunctive satisfaction was introduced in [FKR95]. They present solution for only stationary (memoryless) strategies, and explicitly mention that such strategies are not sufficient and a solution to the general problem would be very useful. They also mention that it is unlikely to be a simple extension of the single dimensional case. Our results not only present the general solution, but we also present results that combine both the conjunctive and joint satisfaction semantics along with the expectation semantics. The multiple percentile are currently considered for various objectives, such as mean-payoff, limsup, liminf, shortest path in [RRS15]. However, [RRS15] does not consider optimizing the expectation, whereas we consider maximizing expectation along with satisfaction semantics. The notion of risks has been considered in MDPs with discounted objectives [WL99], where the goal is to maximize (resp., minimize) the probability (risk) that

the expected total discounted reward (resp., cost) is above (resp., below) a threshold. The notion of strong risk aversion, where for risk the probabilistic choices are treated instead as an adversary was considered in [BFRR14]. In [BFRR14] the problem was considered for single reward for mean-payoff and shortest path. In contrast, though inspired by [BFRR14], we consider risk aversion for multiple reward functions with probabilistic guarantee (instead of adversarial guarantee), which is natural for MDPs. Moreover, [BFRR14] 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 several cases of multiple rewards (see the first item of our contributions). Further, an independent work [CR15] extends [BFRR14] to multiple dimensions, and they also consider "beyond almost-sure threshold problem", which corresponds to the *multi-qual* problem, which is a special case of our solution. Finally, a very different notion of risk has been considered in [BCFK13], where the goal is to optimize the expectation while ensuring low variance. The problem has been considered only for single dimension, and no polynomial-time algorithm is known.

2. Preliminaries

2.1. **Basic definitions.** We mostly follow the basic definitions of [BBC+14] with only minor deviations. We use $\mathbb{N}, \mathbb{Q}, \mathbb{R}$ to denote the sets of positive integers, rational and real numbers, respectively. For $n \in \mathbb{N}$, we denote $[n] = \{1, \ldots, n\}$. For a sequence $\omega = \ell_1 \ell_2 \cdots$ and $n \in \mathbb{N}$, we denote the n-th element by $\omega[n]$.

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 $1 \leq i \leq k$, where $\boldsymbol{v}_i$ denotes the i-th component of vector $\boldsymbol{v}$. Further, $\mathbf{1}$ denotes $(1, \ldots, 1)$, and $\mathbb{1}$ denotes Kronecker's delta, i.e., $\mathbb{1}_x(x) = 1$ and $\mathbb{1}_x(y) = 0$ for $y \neq x$.

Finally, the set of all distributions over a countable set X is denoted by $\mathit{Dist}(X)$, and $d \in \mathit{Dist}(X)$ is Dirac if $d(x) = 1$ for some $x \in X$, i.e., $d = \mathbb{1}_x$.

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 defined as a tuple $G = (S, A, \mathit{Act}, \delta, s_0)$ 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 s_0 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. The semantics of MDPs is defined using the notion of strategies. Intuitively, a strategy in an MDP G is a "recipe" to choose actions. Usually, a strategy is

formally defined as a function $\sigma : (SA)^*S \to \mathit{Dist}(A)$ that given a finite path w, representing the history of a play, gives a probability distribution over the actions enabled in the last state. In this paper, we adopt a slightly different (though equivalent—see [BBC+14, Section 6]) definition, which is more convenient for our setting. Let M be a countable set of *memory elements*. A *strategy* is a triple $\sigma = (\sigma_u, \sigma_n, \alpha)$, where $\sigma_u : A \times S \times \mathsf{M} \to \mathit{Dist}(\mathsf{M})$ and $\sigma_n : S \times \mathsf{M} \to \mathit{Dist}(A)$ are *memory update* and *next move* functions, respectively, and α is 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. $\sigma_n(s, m) \in \mathit{Dist}(\mathit{Act}(s))$.

A *play* of G determined by a strategy σ is a Markov chain $G^\sigma = (S \times \mathsf{M} \times A, P, \mu)$, where

$$\mu(s, m, a) = \mathbb{1}_{s_0}(s) \cdot \alpha(m) \cdot \sigma_n(s, m)(a)$$
$$P(s, m, a)(s', m', a') = \delta(a)(s') \cdot \sigma_u(a, s', m)(m') \cdot \sigma_n(s', m')(a') \,.$$

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 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. "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 $\mathbb{E}^\sigma_s[f] = \int_{\mathsf{Runs}} f \; d\,\mathbb{P}^\sigma_s$ or $\mathbb{E}^\sigma[f] = \int_{\mathsf{Runs}} f \; d\,\mathbb{P}^\sigma$ for short. 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;
- *stochastic-update*, if it is not necessarily deterministic-update;
- *deterministic*, if it is deterministic-update and the next move function gives a Dirac distribution for every argument;
- *randomized*, if it is not necessarily deterministic.

We also classify the strategies according to the size of memory they use. The important subclasses of strategies are

- *memoryless* (or 1-*memory*) strategies, in which M is a singleton,
- n-*memory* strategies, in which M has exactly n elements,
- *finite-memory* strategies, in which M is finite, and
- *Markov* strategies, in which $\mathsf{M} = \mathbb{N}$ and $\sigma_u(\cdot, \cdot, n)(n + 1) = 1$.

Markov strategies have a nice structure: they only need a counter and to know the current state [FV97].

End components. A set $T \cup B$ with $\emptyset \neq T \subseteq S$ and $B \subseteq \bigcup_{t \in T} \mathit{Act}(t)$ is an *end component* of G if (1) for all $a \in B$, whenever $\delta(a)(s') > 0$ then $s' \in T$; and (2) for all $s, t \in T$ there is a path $\omega = s_1 a_1 \cdots a_{k-1} s_k$ 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. Finally, if (S, A) is a MEC, we call the MDP *strongly connected*.

Remark 2.1. The maximal end component (MEC) decomposition of an MDP, i.e., the computation of MEC, can be achieved in polynomial time [CY95]. For improved algorithms for general MDPs and various special cases see [CH11, CH12, CH14, CL13].

Analogously, 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

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

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. [CY98, 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 $\mathbb{P}^\sigma\big[\bigcup_{C\in\mathsf{MEC}} \Omega_C\big] = 1$. Therefore, actions that are not in any MEC are almost surely taken only finitely many times.

2.2. **Problem statement.** In order to define our problem, we first briefly recall how long-run average can be defined. Let $G = (S, A, Act, \delta, s_0)$ 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 point-wise limit inferior:

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

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

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

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

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

We are mostly interested in **(multi-quant-conjunctive)** as it is the core of all other discussed problems. However, we also consider the following important special cases:

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

Additionaly, we are also interested in variants of **(multi-quant-conjunctive)**. Firstly, in **(multi-quant-joint)**, the constraint (conjunctive-SAT) is *replaced* by

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

for $pr \in [0,1]$. Secondly, **(multi-quant-conjunctive-joint)** arises by *adding* (joint-SAT) constraint $\mathbb{P}^\sigma\left[\mathrm{lr}_{\inf}(\boldsymbol{r}) \geq \widetilde{\boldsymbol{sat}}\right] \geq \widetilde{pr}$ for $\widetilde{pr} \in [0,1] \cap \mathbb{Q}$ and $\widetilde{\boldsymbol{sat}} \in \mathbb{Q}^n$. The relationship between the problems is depicted in Fig. 1.

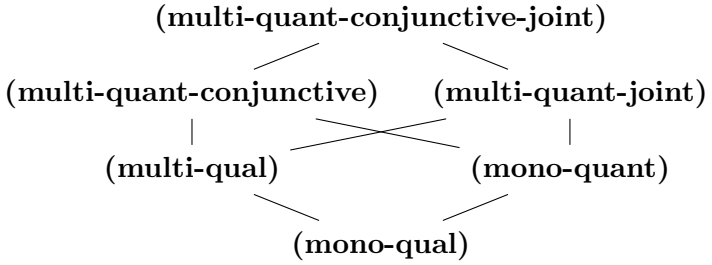

FIGURE 1. Relationship of the defined problems with lower problems being specializations of the higher ones

Furthermore, each of the three constraints (EXP), (conjunctive-SAT), and (joint-SAT) defines the respective decision problem given solely by that constraint. Each of these three problems is a special case of **(multi-quant-conjunctive-joint)** where the other constraints are trivial (e.g. requiring the average reward be greater or equal to the minimum reward of the MDP). Finally, apart from decision problems, one often considers optimization problems, where the task is to maximize the parameters so that the answer to the decision problem is still positive. Observe that since optimization in multi-dimensional setting cannot in general produce a single "best" solution, one can consider Pareto curves, which are sets of all component-wise optimal and mutually incomparable solutions to the optimization problem.

Example 2.2 (Running example)**.** We illustrate **(multi-quant-conjunctive)** with an MDP of Fig. 2 with $n = 2$, rewards as depicted, and $\boldsymbol{exp} = (1.1, 0.5), \boldsymbol{sat} = (0.5, 0.5), \boldsymbol{pr} = (0.8, 0.8)$. 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 $0.8 + 0.8 - 1 = 0.6$ runs have to jointly surpass both satisfaction thresholds (at the same time), i.e. exceed the vector $(0.5, 0.5)$. 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

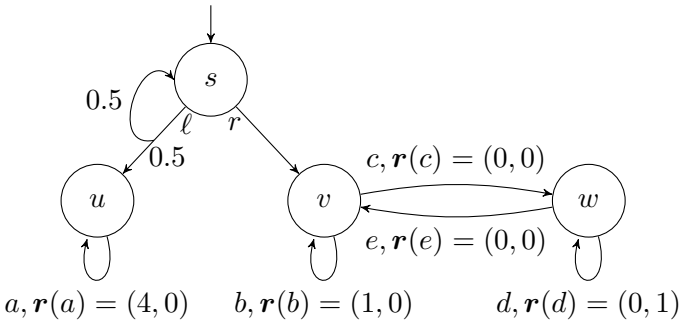

FIGURE 2. An MDP with two-dimensional rewards

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. Also note that in order to do that, we use memory with stochastic update. △

3. 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 the other variants is derived and a detailed analysis of the special cases and the respective complexities is given in Section 6.

3.1. **Previous results.**

3.1.1. *Linear programming for expectation semantics.* In [BBC+14], a solution to 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.

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.

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 $s \in S$

$$\mathbb{1}_{s_0}(s) + \sum_{a \in A} y_a \cdot \delta(a)(s) = \sum_{a \in Act(s)} y_a + y_s$$

(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:

$$\sum_{a \in A} x_a \cdot \boldsymbol{r} \geq \boldsymbol{exp}$$

FIGURE 3. Linear program of [BBC+14] for (EXP)

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 [Put94]. 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.

The solution above builds on the work [EKVY08], which studied MDPs with multiple reachability and ω-regular specifications. It has inspired Equation 1 as well as computation of the Pareto curve. 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 [PY00].

3.1.2. *Linear programming for satisfaction semantics.* Apart from considering (EXP) separately, [BBC+14] also considers the constraint (joint-SAT) separately. While the former was solved using the linear program above, the latter required a reduction to one linear program per each MEC and another one to combine the results. More precisely, for each MEC we first decide whether there is a strategy exceeding the threshold. Second, we maximize the probability to reach these MECs. Similarly, in [RRS15], for each MEC we decide for every subset of thresholds whether there is a strategy exceeding them. The results are again combined in a linear program for reachability.

In contrast, we shall provide a single linear program for the **(multi-quant-conjunctive)** problem, unifying the solution approaches for expectation and satisfaction 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 Section 3.4.

3.2. **Our unifying 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.

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

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

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

Note that x_a represents the absolute frequency of playing a (not relative within the MEC). Intuitively, Equation 6 thus requires in each MEC the average reward be at least $\boldsymbol{sat}$. Here we rely on the non-trivial fact, that in a MEC, actions can be played on almost all runs with the given frequencies for any flow, see Corollary 5.5.

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

3.2.2. *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.

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 3.1. 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.

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 $s \in S$

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

(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 $i \in [n]$

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

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

Example 3.2 (Running example)**.** The linear program L for Example 2.2 is shown in Appendix A. Here we spell out some useful 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.

Moreover, we show later in Lemma 5.1, that 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. △

3.3. **Proof overview.** Here, we briefly describe the main ideas of the proof of Theorem 3.1.

The first point. The complexity follows immediately from the syntax of L and the existence of a polynomial-time algorithm for linear programming [Sch86].

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.

The proof of [BBC$^+$14, 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 [BBC$^+$14], we need to distinguish among runs exceeding various subsets of the value thresholds $\boldsymbol{sat}_i, i \in [n]$. For $N \subseteq [n]$, we call a run N-*good* if $\mathrm{lr}_{\inf}(\boldsymbol{r})_i \geq \boldsymbol{sat}_i$ for exactly 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 3.3 (Running example)**.** The strategy of Example 2.2 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 [EKVY08] yields values satisfying the transient flow equation. Further, similarly as for x-values, instead of simply switching to recurrent behaviour in a particular MEC, we consider switching in a MEC *and* the set N for which the following recurrent behaviour is N-good.

Example 3.4 (Running example)**.** The strategy of Example 2.2 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 2.2). △

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 5. The general pattern follows the proof method of [BBC$^+$14, Proposition 4.5], but there are several important differences.

First, a strategy is designed to behave in a MEC so that the frequencies of actions match the x-values. The structure of the proof differs here and we focus on underpinning the following key principle. 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 [BBC$^+$14], which is convenient for the analysis.

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 3.5 (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 $k-$th step. In Section 5 we discuss the construction of the strategy from the solution in greater detail, necessary for later complexity discussion. △

3.4. **Important aspects of our approach and its consequences.** We now explain some important conceptual aspects of our result. The previous proof idea from [BBC$^+$14] is as follows: (1) The problem for expectation semantics is solved by a linear program. (2) The problem for satisfaction semantics is solved as follows: each MEC is considered, solved separately using a linear program, and then a reachability problem is solved using a different linear program. In comparison, our proof has two conceptual steps. Since our goal is to optimize the expectation (which intuitively requires a linear program), the first step is to come up with a single linear program for 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. $\boldsymbol{w} \cdot \mathrm{lr}_{\inf}(\boldsymbol{r})$, 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 8.

- We can easily add more satisfaction constraints (with different thresholds) on the same resource as well as add joint constraints of the form $\mathbb{P}^\sigma\big[\bigwedge_{k_i} \mathrm{lr}_{\inf}(\boldsymbol{r}_{k_i}) \geq pr\big]$. Both can be solved by adding a copy of Equation 7 for each subset N of all the 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 6. △

4. Proof of Theorem 3.1: Witness strategy induces solution to L

Now we present the technical proof of Theorem 3.1. We start with the second point and show how to construct a solution to L from a witness strategy.

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

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

We construct a solution to the system L. The proof method roughly follows that of [BBC+14, Proposition 4.5]. However, separate flows for "N-good" runs require some careful conditional probability considerations, in particular for Equations 4, 6 and 7.

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

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

Then Ω_N, $N \subseteq [n]$ 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)** from Section 6.

In order to achieve the second point, we define $f_N(a)$, for every a, to be lying between values $\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{P}^\sigma[A_t = a \cap \Omega_N]$ and $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{P}^\sigma[A_t = a \cap \Omega_N]$, which can be safely substituted for $x_{a,N}$ in L. Let A be written as $\{a_1, a_2, \ldots, a_{|A|}\}$ and let us first consider the case when $\mathbb{P}^\sigma[\Omega_N] > 0$. Since every bounded infinite sequence contains an infinite convergent subsequence, there is an increasing sequence of indices, $T_0^1, T_1^1, T_2^1 \ldots$, such that $\lim_{\ell\to\infty} \frac{1}{T_\ell^1}\sum_{t=1}^{T_\ell^1} \mathbb{P}^\sigma[A_t = a_1 \mid \Omega_N]$ is well defined. Then we can choose a subsequence $T_0^2, T_1^2, T_2^2 \ldots$ of the sequence $T_0^1, T_1^1, T_2^1 \ldots$ so that $\lim_{\ell\to\infty} \frac{1}{T_\ell^1}\sum_{t=1}^{T_\ell^1} \mathbb{P}^\sigma[A_t = a_1 \mid \Omega_N]$ is well defined, too. We continue this process for all actions and finally define the sequence $T_0, T_1, T_2 \ldots$ to be $T_0^{|A|}, T_1^{|A|}, T_2^{|A|} \ldots$. Consequently, for each action $a \in A$, the following limit exists

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

and we set for all $a \in A$

$$x_{a,N} \coloneqq f_N(a)$$

Finally, for N such that $\mathbb{P}^\sigma[\Omega_N] = 0$, we set $x_{a,N} := 0$. Note that since actions not in MECs are almost surely taken only finitely many times, we have

$$x_{a,N} = 0 \qquad \text{for } a \notin \bigcup \mathsf{MEC}, N \subseteq [n] \tag{4.1}$$

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 $\delta(a)(s)$. However, we show that if we use the action a with positive frequency then $\Delta_i^N(a)(s)$ approximates $\delta(a)(s)$.

Example 4.1. Consider an action a with $\delta(a)(u) = 0.5$. Then we have $\mathbb{P}^\sigma[S_2 = u \mid A_1 = a] = 0.5$. It may well be that for some set $\Omega \subseteq \mathsf{Runs}$ we have $\mathbb{P}^\sigma[S_2 = u \mid A_1 = a,\ \Omega] = 1$, but then $\mathbb{P}^\sigma[\Omega] \leq 0.5$. Similarly, if $\mathbb{P}^\sigma[S_2 = u \mid A_1 = a,\ \Omega] = \mathbb{P}^\sigma[S_3 = u \mid A_2 = a,\ \Omega] = 1$ then $\mathbb{P}^\sigma[\Omega] \leq 0.25$, and so on. In general, whenever $\mathbb{P}^\sigma[\Omega] > 0$, the transition probabilities on Ω cannot differ from the actual transition probabilities too much all the time. △

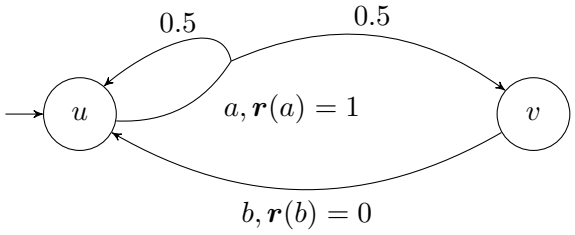

FIGURE 5. An MDP illustrating Δ

We first consider a simpler problem:

Lemma 4.2. Let $(\Delta_t)_{t\in\mathbb{N}}$ be i.i.d. Bernoulli variables with expectation $\delta = \mathbb{E}[\Delta_t]$. Then for any event Ω with $\mathbb{P}[\Omega] > 0$, we have $\lim_{t\to\infty} \mathbb{E}_\Omega[\Delta_t] = \delta$.

Proof. For a contradiction, let w.l.o.g. $\limsup_{t\to\infty} \mathbb{E}_\Omega[\Delta_t] = \delta + 3\varepsilon$. (If $\limsup_{t\to\infty} \mathbb{E}_\Omega[\Delta_t] < \delta$, we can consider the variables $1 - \Delta_t$ with this property). Moreover, we may safely assume that $\mathbb{E}_\Omega[\Delta_t] \geq \delta + 2\varepsilon$ for all $t \in \mathbb{N}$, otherwise we consider the respective subsequence. Let $\mathit{High}_i \subseteq \Omega$ be the set of runs of Ω such that $\frac{1}{i}\sum_{t=1}^{i} \Delta_t > \delta + \varepsilon$ and similarly $\mathit{Normal}_i \subseteq \Omega$ be the set of runs of Ω such that $\frac{1}{i}\sum_{t=1}^{i} \Delta_t \leq \delta + \varepsilon$. Clearly, $\Omega = \mathit{High}_i \uplus \mathit{Normal}_i$ for every i. Then

$$\begin{aligned}\delta + 2\varepsilon \leq \frac{1}{i}\sum_{t=1}^{i} \mathbb{E}_\Omega[\Delta_t] &= \frac{1}{i}\mathbb{E}_\Omega\left[\sum_{t=1}^{i} \Delta_t\right] \\ &= \frac{\frac{1}{i}\mathbb{E}_{\mathit{High}_i}[\sum_{t=1}^{i} \Delta_t] \cdot \mathbb{P}[\mathit{High}_i] + \frac{1}{i}\mathbb{E}_{\mathit{Normal}_i}[\sum_{t=1}^{i} \Delta_t] \cdot \mathbb{P}[\mathit{Normal}_i]}{\mathbb{P}[\mathit{High}_i] + \mathbb{P}[\mathit{Normal}_i]}\end{aligned}$$

$$\leq \frac{1 \cdot \mathbb{P}[High_i] + (\delta + \varepsilon) \cdot \mathbb{P}[Normal_i]}{\mathbb{P}[High_i] + \mathbb{P}[Normal_i]}$$

Altogether, by comparing the first and the last expression, we get

$$\mathbb{P}[Normal_i] \leq \frac{1 - \delta - 2\varepsilon}{\varepsilon} \cdot \mathbb{P}[High_i] \tag{4.2}$$

where the fraction is constant for all i. Since by the law of large numbers $\lim_{i\to\infty} \mathbb{P}[High_i] = 0$, we obtain $\lim_{i\to\infty} \mathbb{P}[Normal_i] = 0$ and thus $\mathbb{P}[\Omega] = 0$, a contradiction. □

Now we apply the preceding lemma to MDPs:

Lemma 4.3. Let $N \subseteq [n]$ be such that $\mathbb{P}^\sigma[\Omega_N] > 0$. Then for every $a \in A, s \in S$, we have $\lim_{t\to\infty} \mathbb{P}^\sigma[A_t = a \mid \Omega_N] \cdot |\Delta_t^N(a)(s) - \delta(a)(s)| = 0$.

Proof plan. Note that if $\mathbb{P}^\sigma[A_t = a \mid \Omega_N] = 1$ for all t then the result follows directly from the previous lemma where we set $\Delta_t(\omega)$ to 1 if $S_{t+1} = s$ and 0 otherwise. Indeed, then $\mathbb{E}[\Delta_t] = \delta(a)(s)$ and $\mathbb{E}_{\Omega_N}[\Delta_t] = \Delta_t^N(a)(s)$. Consequently, $\lim_{t\to\infty} \mathbb{P}^\sigma[A_t = a \mid \Omega_N] \cdot |\Delta_t^N(a)(s) - \delta(a)(s)| = 1 \cdot 0$.

In the general case, the probability of taking a on the runs can vary over time. In order to cope with that, we consider sets $I \subset \mathbb{N}$ of positions where a is taken with high enough probability (i.e., in "many" runs). The first step of the proof is thus to derive (4.3), an analogue of (4.2), but now relativized to positions in I. In the previous lemma, the second step consisted in applying the law of large numbers to conclude that probability of overly high preference of some outcome has zero probability, causing a contradiction with (4.2). In this proof, the second step will require more math to conclude that, due to the relativization.

Proof. Suppose for a contradiction, that for some $a \in A, s \in S$ there are infinitely many t for which $\mathbb{P}^\sigma[A_t = a \mid \Omega_N] \cdot |\Delta_t^N(a)(s) - \delta(a)(s)| > \xi$ for some $\xi > 0$. Denote the set of these t's by T. Since both factors are bounded by 0 and 1, there are $\zeta > 0$ and $\varepsilon > 0$ such that for all $t \in T$ we have $\mathbb{P}^\sigma[A_t = a \mid \Omega_N] > \zeta$ and w.l.o.g. $\Delta_t^N(a)(s) > \delta(a)(s) + 2\varepsilon$ (if $\Delta_t^N(a)(s) < \delta(a)(s)$ then there is another successor s' of a with this property). Consequently, for every $t \in T$, we have

$$\frac{\mathbb{P}^\sigma[\Omega_N \cap A_t = a \cap S_{t+1} = s]}{\mathbb{P}^\sigma[\Omega_N \cap A_t = a]} > \delta(a)(s) + 2\varepsilon$$

First step. Now we derive (4.3), a version of (4.2) relativized to finite sets $I \subseteq T$. The positive probability of taking a in these positions guarantees that overly high preference of the outcome s is well defined.

Formally, similarly to the previous inequality for each $t \in T$, the same holds for the average over any finite set of indices $I \subseteq T$:

$$\delta(a)(s) + 2\varepsilon < \frac{\sum_{t\in I} \mathbb{P}^\sigma[\Omega_N \cap A_t = a \cap S_{t+1} = s]}{\sum_{t\in I} \mathbb{P}^\sigma[\Omega_N \cap A_t = a]} = (*)$$

Denoting

$$i\text{-Tries-In-}I = \{\omega \in \Omega_N \mid |\{t \in I \mid A_t = a\}| = i\}$$
$$i\text{-Successes-In-}I = \{\omega \in \Omega_N \mid |\{t \in I \mid A_t = a \cap S_{t+1} = s\}| = i\}$$

we can rewrite the term $(*)$ by grouping runs with same "frequencies" as

$$(*) = \frac{\sum_{i=1}^{|I|} i \cdot \mathbb{P}^\sigma[i\text{-Successes-In-}I]}{\sum_{i=1}^{|I|} i \cdot \mathbb{P}^\sigma[i\text{-Tries-In-}I]} = (**)$$

Similarly to the previous lemma, we introduce runs with "success rate" higher and lower than $\delta(a)(s) + \varepsilon$, now relative to the indices of I. Formally,

$$\mathit{High}_i^I = i\text{-Tries-In-}I \cap \bigcup_{k > i\cdot\left(\delta(a)(s)+\varepsilon\right)} k\text{-Successes-In-}I$$

$$\mathit{Normal}_i^I = i\text{-Tries-In-}I \cap \bigcup_{k \leq i\cdot\left(\delta(a)(s)+\varepsilon\right)} k\text{-Successes-In-}I$$

allows us to rewrite

$$(**) = \frac{\sum_{i=1}^{|I|}(i \cdot \mathit{HighRate}_i) \cdot \mathbb{P}^\sigma\left[\mathit{High}_i^I\right] + \sum_{i=1}^{|I|}(i \cdot \mathit{NormalRate}_i) \cdot \mathbb{P}^\sigma\left[\mathit{Normal}_i^I\right]}{\sum_{i=1}^{|I|} i \cdot \mathbb{P}^\sigma\left[\mathit{High}_i^I\right] + \sum_{i=1}^{|I|} i \cdot \mathbb{P}^\sigma\left[\mathit{Normal}_i^I\right]} = (***)$$

where each $\mathit{HighRate}_i \in (\delta(a)(s) + \varepsilon, 1]$ and $\mathit{NormalRate}_i \in [0, \delta(a)(s) + \varepsilon]$ are the average portions of "successes" among the "tries" in the respective High_i^I and Normal_i^I. Hence we can safely use the upper bounds to show

$$(***) \leq \frac{1 \cdot \sum_{i=1}^{|I|} i \cdot \mathbb{P}^\sigma\left[\mathit{High}_i^I\right] + (\delta(a)(s) + \varepsilon) \cdot \sum_{i=1}^{|I|} i \cdot \mathbb{P}^\sigma\left[\mathit{Normal}_i^I\right]}{\sum_{i=1}^{|I|} i \cdot \mathbb{P}^\sigma\left[\mathit{High}_i^I\right] + \sum_{i=1}^{|I|} i \cdot \mathbb{P}^\sigma\left[\mathit{Normal}_i^I\right]} = (****)$$

Since $(****) \geq (*) \geq \delta(a)(s) + 2\varepsilon$, we get by the same computation as for obtaining (4.2)

$$\sum_{i=1}^{|I|} i \cdot \mathbb{P}^\sigma\left[\mathit{Normal}_i^I\right] \leq \frac{1 - \delta - 2\varepsilon}{\varepsilon} \cdot \sum_{i=1}^{|I|} i \cdot \mathbb{P}^\sigma\left[\mathit{High}_i^I\right] \tag{4.3}$$

for every finite $I \subseteq T$.

Second step. Now we consider particular I's leading to a contradiction. Let T be written as $\{t_1, t_2, \ldots\}$ so that $t_1 < t_2 < \cdots$. For $m < n$, we consider finite subsets $I_m^n = \{t_m, t_{m+1}, \ldots, t_n\}$ of T and will prove that

$$\lim_{m\to\infty} \lim_{n\to\infty} \sum_{i=1}^{|I_m^n|} i \cdot \mathbb{P}^\sigma\left[\mathit{High}_i^{I_m^n}\right] = 0 \tag{4.4}$$

As a consequence of (4.3) we obtain also $\lim_{m\to\infty} \lim_{n\to\infty} \sum_{i=1}^{|I_m^n|} i \cdot \mathbb{P}^\sigma\left[\mathit{Normal}_i^{I_m^n}\right] = 0$ and thus $\lim_{m\to\infty} \lim_{n\to\infty} \sum_{i=1}^{|I_m^n|} i \cdot \mathbb{P}^\sigma[i\text{-Tries-In-}I_m^n] = 0$, i.e. with growing m the average number of tries after m approaches 0, a contradiction with $\mathbb{P}^\sigma[A_t = a \mid \Omega_N] > \zeta$ for infinitely many t and $\mathbb{P}^\sigma[\Omega_N] > 0$.

It remains to prove (4.4). Intuitively, we consider index sets that start later (at position $m \to \infty$) to avoid initial potentially large elements. Summands with high i's, i.e. runs with many tries, below denoted by $\mathcal{C}$, will be shown negligible by the central limit theorem (in the previous lemma the law of large numbers was sufficient). Further, we will have to argue that even summands with low i's are small for high enough m. This is due to the fact that either a is taken frequently enough on some runs ($\mathcal{A}$) or for high enough indices not any more on the other runs ($\mathcal{B}$).

Formally, let $Inf = \Omega_N \cap \{A_t = a \text{ for infinitely many } t\}$ and $Fin_{\geq k} = \Omega_N \cap \{A_t = a \text{ for only finitely many } t\} \cap \{A_t = a \text{ for some } t \geq k\}$. We split the sum $\sum_{i=1}^{|I_m^n|} i \cdot \mathbb{P}^\sigma\left[High_i^{I_m^n}\right]$ into

$$\underbrace{\sum_{i=1}^{middle(m)} i \cdot \mathbb{P}^\sigma\left[High_i^{I_m^n} \cap Inf\right]}_{\mathcal{A}} + \underbrace{\sum_{i=1}^{middle(m)} i \cdot \mathbb{P}^\sigma\left[High_i^{I_m^n} \cap Fin_{\geq m}\right]}_{\mathcal{B}} + \underbrace{\sum_{i=middle(m)+1}^{|I_m^n|} i \cdot \mathbb{P}^\sigma\left[High_i^{I_m^n}\right]}_{\mathcal{C}}$$

by defining an appropriate $middle : \mathbb{N} \to \mathbb{N}$. We show that each term approaches zero.

- $\mathcal{A}$: Observe that for every i and m, we have $\lim_{n\to\infty} \mathbb{P}^\sigma[i\text{-Tries-In-}I_m^n \cap Inf] = 0$. Hence also $\lim_{n\to\infty} \mathcal{A} = 0$ for every m and irrespective of the choice of $middle(m)$, and thus $\lim_{m\to\infty} \lim_{n\to\infty} \mathcal{A} = 0$.
- $\mathcal{B}$: We define $middle(m)$ to be the largest number such that $\sum_{i=1}^{middle(m)} i \cdot \mathbb{P}^\sigma[Fin_{\geq m}] < 1/m$. This trivially ensures $\lim_{m\to\infty} \mathcal{B} \leq \lim_{m\to\infty} 1/m = 0$.
- $\mathcal{C}$: Since $\lim_{m\to\infty} \mathbb{P}^\sigma[Fin_{\geq m}] = 0$, we obtain by the definition of $middle$ that for $m \to \infty$ also $middle(m) \to \infty$. Consequently, it is sufficient to prove that

$$\lim_{n\to\infty} \sum_{i=k}^{|I_m^n|} i \cdot \mathbb{P}^\sigma\left[High_i^{I_m^n}\right] \to 0 \text{ for } k \to \infty \text{ uniformly for all } m\,. \tag{4.5}$$

Fix an arbitrary m. Let X_j denote the indicator random variable of the event that jth use of action a, when looking only at time points $t_m, t_{m+1}, t_{m+2} \ldots$, resulted in the successor s. Precisely, let T_j be an auxiliary random variable with value t_ℓ such that $|\{q \mid m \leq q \leq \ell, A_{t_q} = a\}| = j$ and $A_{t_q} = a$; then X_j is 1 if $S_{T_j+1} = s$ and 0 otherwise. Due to the Markov property, X_j are Bernoulli i.i.d. with mean $\delta(a)(s)$. Further,

$$High_i^{I_m^n} \subseteq \left\{ \frac{\sum_{j=1}^{i} X_j}{i} > \delta(a)(s) + \varepsilon \right\}$$

Therefore, by central limit theorem

$$\mathbb{P}^\sigma\left[High_i^I\right] \lessapprox \Phi(-\sqrt{i} \cdot \hat{\varepsilon})$$

where $\hat{\varepsilon} = \varepsilon / \sqrt{\delta(a)(s) \cdot (1 - \delta(a)(s))}$ and Φ is the cumulative distribution function of the standard normal distribution and $\lessapprox$ denotes that the inequality $\leq$ holds "only for large i", i.e. in the limit. Consequently, for large k, we have

$$\lim_{n\to\infty} \sum_{i=k}^{|I_m^n|} i \cdot \mathbb{P}^\sigma\left[High_i^{I_m^n}\right] \lessapprox \sum_{i=k}^{\infty} i \cdot \Phi(-\sqrt{i} \cdot \hat{\varepsilon})$$

where the right-hand side does not depend on m and is thus a uniform bound for all m. Further, since $\Phi(-\sqrt{i} \cdot \hat{\varepsilon})$ decreases exponentially in $\sqrt{i}$, the right-hand side approaches 0 as $k \to 0$ (independently of m) and (4.5) follows. □

Now we show, that Equation 4 is satisfied. For all $s \in S$ and $N \subseteq [n]$ such that $\mathbb{P}^\sigma[\Omega_N] = 0$, we have trivially

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

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

$$\begin{aligned}
&\frac{1}{\mathbb{P}^\sigma[\Omega_N]} \sum_{a \in A} f_N(a) \cdot \delta(a)(s) \\
&= \frac{1}{\mathbb{P}^\sigma[\Omega_N]} \sum_{a \in A} \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\sigma[A_t = a \mid \Omega_N] \cdot \mathbb{P}^\sigma[\Omega_N] \cdot \delta(a)(s) && \text{(definition of } f_N\text{)} \\
&= \sum_{a \in A} \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\sigma[A_t = a \mid \Omega_N] \cdot \delta(a)(s) && \text{(linearity of the limit)} \\
&= \sum_{a \in A} \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\sigma[A_t = a \mid \Omega_N] \cdot \Delta_t^N(a)(s) && \text{(Lemma 4.3)} \\
&= \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \sum_{a \in A} \mathbb{P}^\sigma[A_t = a \mid \Omega_N] \cdot \Delta_t^N(a)(s) && \text{(definition of } T_\ell\text{)} \\
&= \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\sigma[S_{t+1} = s \mid \Omega_N] && \text{(definition of } \Delta_t^N\text{)} \\
&= \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\sigma[S_t = s \mid \Omega_N] && \text{(reindexing and Cesaro limit)} \\
&= \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \sum_{a \in Act(s)} \mathbb{P}^\sigma[A_t = a \mid \Omega_N] && (s \text{ must be followed by } a \in Act(s)) \\
&= \frac{1}{\mathbb{P}^\sigma[\Omega_N]} \sum_{a \in Act(s)} \lim_{\ell \to \infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\sigma[A_t = a \mid \Omega_N] \cdot \mathbb{P}^\sigma[\Omega_N] && \text{(linearity of the limit)} \\
&= \frac{1}{\mathbb{P}^\sigma[\Omega_N]} \sum_{a \in Act(s)} f_N(a)\,. && \text{(definition of } f_N\text{)}
\end{aligned}$$

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

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

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

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

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

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

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

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

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

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

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

Although Fatou's lemma (see, e.g. [Roy88, 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.

In order to show that Equations 6 and 7 hold, we prove the following lemma. This lemma is further necessary when relating the x-variables to the transient flow in Equation 3 later.

Lemma 4.4. 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. The proof is trivial for the case with $\mathbb{P}^\sigma[\Omega_N]=0$. Let us now assume $\mathbb{P}^\sigma[\Omega_N]>0$:

$$\sum_{a\in C} x_{a,N}$$

$$= \lim_{\ell\to\infty}\frac{1}{T_\ell}\sum_{t=1}^{T_\ell}\sum_{a\in C}\mathbb{P}^\sigma[A_t=a\mid\Omega_N]\cdot\mathbb{P}^\sigma[\Omega_N] \qquad \text{(definition of } x_{a,N} \text{ and } T_\ell\text{)}$$

$$= \lim_{\ell\to\infty}\frac{1}{T_\ell}\sum_{t=1}^{T_\ell}\sum_{a\in C}\Big(\mathbb{P}^\sigma[A_t=a\mid\Omega_N\cap\Omega_C]\cdot\frac{\mathbb{P}^\sigma[\Omega_N\cap\Omega_C]}{\mathbb{P}^\sigma[\Omega_N]}+$$

$$\mathbb{P}^\sigma[A_t=a\mid\Omega_N\setminus\Omega_C]\cdot\frac{\mathbb{P}^\sigma[\Omega_N\setminus\Omega_C]}{\mathbb{P}^\sigma[\Omega_N]}\Big)\cdot\mathbb{P}^\sigma[\Omega_N] \qquad \text{(partitioning of } \mathsf{Runs}\text{)}$$

$$= \lim_{\ell\to\infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \sum_{a\in C} \mathbb{P}^\sigma[A_t = a \mid \Omega_N \cap \Omega_C] \cdot \mathbb{P}^\sigma[\Omega_N \cap \Omega_C]$$

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

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

$$= \mathbb{P}^\sigma[\Omega_N \cap \Omega_C] \cdot \lim_{\ell\to\infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\sigma[A_t \in C \mid \Omega_N \cap \Omega_C]$$

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

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

It remains to prove that the last limit is equal to 1. We have

$$1 \geq \lim_{\ell\to\infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{P}^\sigma[A_t \in C \mid \Omega_N \cap \Omega_C] = \lim_{\ell\to\infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \mathbb{E}^\sigma\left[\sum_{a\in C} \mathbb{1}_a(A_t) \mid \Omega_N \cap \Omega_C\right]$$

which is by dominated convergence theorem equal to

$$\mathbb{E}^\sigma\left[\lim_{\ell\to\infty} \frac{1}{T_\ell} \sum_{t=1}^{T_\ell} \sum_{a\in C} \mathbb{1}_a(A_t) \mid \Omega_N \cap \Omega_C\right] = \mathbb{E}^\sigma[1] = 1$$

by definition of Ω_C. □

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 for Eq. 5, by def. of $x_{a,N}, f_N$, linearity of lim, def. of lim inf)

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

(as above in Lemma 4.4, by partitioning Runs, now with additional factor $\boldsymbol{r}_i(a)$)

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

(as above for Eq. 5, by def. of expectation and Fatou's lemma)

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

$$= \sum_{a\in C} x_{a,N} \cdot \boldsymbol{sat}_i \qquad \text{(by Lemma 4.4)}$$

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

$$\begin{aligned}
\sum_{N \subseteq [n]: i \in N} \sum_{a \in A} x_{a,N} &= \sum_{N \subseteq [n]: i \in N} \sum_{C \in \mathsf{MEC}} \sum_{a \in C} x_{a,N} && \text{(by (4.1))} \\
&= \sum_{N \subseteq [n]: i \in N} \sum_{C \in \mathsf{MEC}} \mathbb{P}^\sigma[\Omega_N \cap \Omega_C] && \text{(by Lemma 4.4)} \\
&= \sum_{N \subseteq [n]: i \in N} \mathbb{P}^\sigma[\Omega_N] && (\Omega_C\text{'s partition almost all } \mathsf{Runs})
\end{aligned}$$

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

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

Lemma 4.5. There is a strategy $\overline{\sigma}$ in $\overline{G}$ such that for every $C \in \mathsf{MEC}$ and $N \subseteq [n]$,

$$\sum_{s \in C} \mathbb{P}^{\overline{\sigma}}_{\overline{s_0}}[\Diamond f_{s,N}] = \mathbb{P}^\sigma_{s_0}[\Omega_C \cap \Omega_N] \ .$$

Proof. First, we consider an MDP G' created from G in the same way as $\overline{G}$, but instead of $f_{s,N}$ for each $s \in S, N \subseteq [n]$, we only have a single f_s; similarly for actions a_s. As in [BBC+14, Lemma 4.6], we obtain a strategy σ' in G' such that $\sum_{s \in C} \mathbb{P}^{\sigma'}_{s'_0}[\Diamond f_s] = \mathbb{P}^\sigma_{s_0}[\Omega_C]$. We modify σ' into $\overline{\sigma}$ as follows. It behaves as σ', but instead of taking action a_s with probability p, we take each action $a_{s,N}$ with probability $p \cdot \frac{\mathbb{P}^\sigma_{s_0}[\Omega_C \cap \Omega_N]}{\mathbb{P}^\sigma_{s_0}[\Omega_C]}$. (For $\mathbb{P}^\sigma_{s_0}[\Omega_C] = 0$, we define $\overline{\sigma}$ arbitrarily.) Then

$$\sum_{s \in C} \mathbb{P}^{\overline{\sigma}}_{\overline{s_0}}[\Diamond f_{s,N}] = \sum_{s \in C} \frac{\mathbb{P}^\sigma_{s_0}[\Omega_C \cap \Omega_N]}{\mathbb{P}^\sigma_{s_0}[\Omega_C]} \cdot \mathbb{P}^{\sigma'}_{s'_0}[\Diamond f_s] = \mathbb{P}^\sigma_{s_0}[\Omega_C \cap \Omega_N] \qquad \square$$

By [EKVY08, Theorem 3.2], there is a memoryless strategy $\overline{\sigma}$ satisfying the lemma above such that

$$y_a := \sum_{t=1}^{\infty} \mathbb{P}^{\overline{\sigma}}_{\overline{s}}[A_t = a] \qquad \text{(for actions } a \text{ preserved in } \overline{G}\text{)}$$
$$y_{s,N} := \mathbb{P}^{\overline{\sigma}}_{\overline{s_0}}[\Diamond f_{s,N}]$$

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

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

By Lemma 4.5 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 Lemma 4.4, yielding Equation 3.

5. Proof of Theorem 3.1: Solution to L induces witness strategy

Now we proceed to the proof of the third point of Theorem 3.1. Let $x_{a,N}, y_a, y_{s,N}, s \in S, a \in A, N \subseteq [n]$ be a solution to the system L. We show how it effectively induces a witness strategy σ.

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, recall $\mathbb{1}_a$ is 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. Secondly, we eliminate this error as we let the transiting happen with measure vanishing over time.

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

Lemma 5.1. Let $x_{a,N}, a \in A, N \subseteq [n]$ be a non-negative solution to Equation 4 of system L. Then for any fixed N, $X_N := \{s, a \mid x_{a,N} > 0, a \in Act(s)\}$ 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 $x_{a,N} = 0$.

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

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

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

Secondly, let there be a path from $\hat{s}$ to $\hat{t}$ in X_N. We need to show that there is a path from $\hat{t}$ to $\hat{s}$ in X_N. Assume the contrary and denote $T \subseteq X_N$ the set of states with no path to $\hat{s}$ in X_N; we assume $\hat{t} \in T$. We write the path from $\hat{s}$ to $\hat{t}$ as $\hat{s} \cdots s' b t' \cdots \hat{t}$ where $s' \in X_N \setminus T$ and $t' \in T$. Then $b \in Act(s')$ and $\delta(b)(t') > 0$. Consequently,

$$\begin{aligned}
\sum_{s \in X_N \setminus T} \sum_{a \in A} x_a \cdot \delta(a)(s) &= \sum_{s \in X_N \setminus T} \sum_{a \in Act(s)} x_a && \text{(by summing Equation 4 over } s \in X_N \setminus T) \\
&= \sum_{s \in X_N \setminus T} \sum_{a \in Act(s)} \sum_{\overline{s} \in X_N \setminus T} x_a \cdot \delta(a)(\overline{s}) + \sum_{s \in X_N \setminus T} \sum_{a \in Act(s)} \sum_{\overline{s} \in T} x_a \cdot \delta(a)(\overline{s}) \\
& && \text{(case split over target states)} \\
&> \sum_{s \in X_N \setminus T} \sum_{a \in Act(s)} \sum_{\overline{s} \in X_N \setminus T} x_a \cdot \delta(a)(\overline{s}) && \text{(by } \delta(b)(t') > 0) \\
&= \sum_{\overline{s} \in X_N \setminus T} \sum_{\substack{a \in Act(s): \\ s \in X_N \setminus T}} x_a \cdot \delta(a)(\overline{s}) && \text{(rearranging)} \\
&= \sum_{\overline{s} \in X_N \setminus T} \sum_{a \in A} x_a \cdot \delta(a)(\overline{s}) && \text{(see below)}
\end{aligned}$$

which is a contradiction. The last equality follows by definition of T: actions enabled in T cannot lead to $X_N \setminus T$ since from $X_N \setminus T$ there is always a path to $\hat{s}$ and from T there is no path to $\hat{s}$. □

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

$$\sum_{a \in A} x_a \cdot \delta(a)(s) = \sum_{a \in Act(s)} x_a \qquad \text{for each } s \in S. \tag{5.1}$$

We set out to prove Corollary 5.5. 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 x_a (each end-component of X_N of Lemma 5.1).

Lemma 5.2. In a strongly connected MDP G, 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 $\sum_{a\in A} x_{a,N} > 0$. 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 holds

$$\boldsymbol{Freq}_a = \frac{x_{a,N}}{\sum_{a\in D\cap A} x_{a,N}}$$

i.e. $\mathbb{P}^\zeta\left[\boldsymbol{Freq}_a = \frac{x_{a,N}}{\sum_{a\in D\cap A} x_{a,N}} \mid \Omega_D\right] = 1$. Moreover, if all $x_{a,N}$'s are positive then G^ζ is a BSCC and $\boldsymbol{Freq}_a$ is almost surely constant.

Proof. By [BBC+14, Lemma 4.3] applied to Equation (5.1), we get a memoryless strategy ζ such that $\mathbb{E}^\zeta[\boldsymbol{Freq}_a \mid \Omega_D] = x_{a,N}/\sum_{a\in D\cap A} x_{a,N}$. Furthermore, by the ergodic theorem, $\boldsymbol{Freq}_a$ returns the same value for almost all runs in Ω_D, hence is equal to $\mathbb{E}^\zeta[\boldsymbol{Freq}_a \mid \Omega_D]$. Finally, if all $x_{a,N}$'s are positive then all actions of G are used. Consequently, since G is strongly connected, G^ζ is also strongly connected. □

Secondly, we connect the parts (more end components of Lemma 5.1 within one MEC) and thus average the frequencies. This happens at a cost of small error used for transiting between the strongly connected parts.

Lemma 5.3. 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 memoryless 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 [BBC+14, Proposition 5.1, Part 2].

There exists an arbitrarily small (strictly) positive solution x'_a of Equation (5.1). 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 (5.1) is linear and homogeneous, assigning $x_{a,N} + x'_a$ to $x_{a,N}$ also solves this system (and thus Equation 4 as well) and all values are positive. Consequently, Lemma 5.2 gives us a memoryless strategy ζ^ε satisfying almost surely (with $\mathbb{P}^{\zeta^\varepsilon}$-probability 1)

$$\boldsymbol{Freq}_a = \frac{(x_{a,N} + x'_a)}{\sum_{a'\in A}\left(x_{a',N} + x'_{a'}\right)}.$$

We may safely assume that $\sum_{a\in A} x'_a \le \frac{\varepsilon}{1-\varepsilon} \cdot \sum_{a\in A} x_{a,N}$. Then almost surely

$$\begin{aligned}
\boldsymbol{Freq}_a &= \frac{x_{a,N} + x'_a}{\sum_{a\in A}(x_{a,N} + x'_a)} && \text{(by Lemma 5.2)}\\
&> \frac{x_{a,N}}{\sum_{a\in A} x_{a,N} + \sum_{a\in A} x'_a} && \text{(by } x'_a > 0\text{)}\\
&\ge \frac{x_{a,N}}{\sum_{a\in A} x_{a,N} + \frac{\varepsilon}{1-\varepsilon} \cdot \sum_{a\in A} x_{a,N}} && \text{(by } \textstyle\sum_{a\in A} x'_a \le \frac{\varepsilon}{1-\varepsilon} \cdot \sum_{a\in A} x_{a,N}\text{)}\\
&= \frac{x_{a,N}}{\frac{1}{1-\varepsilon} \cdot \sum_{a\in A} x_{a,N}} && \text{(rearranging)}
\end{aligned}$$

$$= \frac{x_{a,N}}{\sum_{a\in A} x_{a,N}} - \varepsilon \cdot \frac{x_{a,N}}{\sum_{a\in A} x_{a,N}} \qquad \text{(rearranging)}$$

$$\geq \frac{x_{a,N}}{\sum_{a\in A} x_{a,N}} - \varepsilon \qquad \text{(by } \tfrac{x_{a,N}}{\sum_{a\in A} x_{a,N}} \leq 1\text{)}$$

□

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

Lemma 5.4. 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 Markov strategy ξ such that almost surely

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

Proof. This proof very closely follows the computation in [BBC+14, Proposition 5.1, Part "Moreover"], but for general ξ_i.

Given $a \in A$, let $lf_a := \lim_{i\to\infty} \boldsymbol{f}^i_a$. 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 $\mathbb{P}^{\xi_j}\left[\mathrm{lr}_{\inf}(\mathbb{1}_a) \geq lf_a - 2^{-j-1}\right] = 1$. 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

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

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

$$\frac{\sum_{k<j} n_k}{n_j} \leq 2^{-j} \tag{5.2}$$

$$\frac{\kappa_{j+1}}{n_j} \leq 2^{-j} \tag{5.3}$$

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). Note that such strategy is a Markov strategy.

Let us give some intuition behind ξ. The numbers in the sequence $n_0, n_1, \ldots$ grow rapidly so that after ξ_j is simulated for n_j steps, the part of the history when ξ_k for $k < j$ were simulated becomes relatively small and has only minor impact on the current average reward (this is ensured by the condition $\frac{\sum_{k<j} n_k}{n_j} \leq 2^{-j}$). This gives us that almost every run has infinitely many prefixes on which the average reward w.r.t. $\mathbb{1}_a$ is arbitrarily close to lf_a infinitely often. To get that lf_a is also the long-run 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

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

Denote by E_k the set of all runs $\omega = s_0a_0s_1a_1\cdots$ of G^ξ such that for some $\kappa_k \leq d \leq n_k$ we have

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

We have $\mathbb{P}^\xi[E_j] \leq 2^{-j}$ and thus $\sum_{j=1}^{\infty}\mathbb{P}^\xi[E_j] = \frac{1}{2} < \infty$ holds. By the Borel-Cantelli lemma [Roy88], almost surely only finitely many of E_j take place. Thus, almost every run $\omega = s_0a_0s_1a_1\cdots$ 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

$$\frac{1}{d}\sum_{k=N_j}^{N_j+d-1} \mathbb{1}_a(a_k) \quad \geq \quad lf_a - 2^{-j}\,. \tag{5.4}$$

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

$$\frac{1}{T}\sum_{t=0}^{T} \mathbb{1}_a(a_t) \quad \geq \quad (lf_a - 2^{1-j})(1-2^{1-j})\,. \tag{5.5}$$

Taking the limit of (5.5) where T (and thus also j) goes to ∞, we obtain

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

yielding the lemma. It remains to prove (5.5). 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 (5.4)

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

which gives

$$\frac{1}{T}\sum_{t=0}^{T} \mathbb{1}_a(a_t) \;\geq\; (lf_a - 2^{1-j})\frac{n_j}{T} \;+\; \frac{1}{T}\sum_{t=N_j}^{T} \mathbb{1}_a(a_t). \tag{5.6}$$

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 (5.2) and (5.3). Therefore, by (5.6),

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

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

$$\begin{aligned}
\frac{1}{T}\sum_{t=N_j}^{T} \mathbb{1}_a(a_t) &= \frac{1}{T-N_j+1}\sum_{t=N_j}^{T} \mathbb{1}_a(a_t) \cdot \frac{T-N_j+1}{T} \\
&\geq (lf_a - 2^{-j})\left(1 - \frac{N_{j-1}+n_j}{T}\right) && \text{(by (5.4))} \\
&\geq (lf_a - 2^{-j})\left(1 - 2^{-j} - \frac{n_j}{T}\right) && \text{(by (5.2))}
\end{aligned}$$

and thus, by (5.6),

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

which finishes the proof of (5.5). □

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 5.5. 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 Markov 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 Lemma 5.4 taking ξ_i to be $\zeta^{1/i}$ from Lemma 5.3. □

Remark 5.6. Note that using such strategy, all actions and states in the single MEC are visited infinitely often. (This will be later useful for the strategy complexity analysis.)

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

$$\mathrm{lr}_{\inf}(\boldsymbol{r})(\omega) = \frac{\sum_a x_{a,N} \cdot \boldsymbol{r}(a)}{\sum_a x_{a,N}}\,.$$

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

$$\mathbb{E}^{\xi_N}[\mathrm{lr}_{\inf}(\boldsymbol{r}) \mid \Omega_C] = \frac{\sum_{a\in C\cap A} x_{a,N} \cdot \boldsymbol{r}(a)}{\sum_{a\in C\cap A} x_{a,N}}\,. \tag{5.7}$$

Moreover, the long-run average reward is the same for almost all runs, which is a stronger property than in [BBC+14, Lemma 4.3], which does not hold for the induced strategy there. We need this property here in order to combine the satisfaction requirements.

$$\mathbb{P}^{\xi_N}\left[\mathrm{lr}_{\inf}(\boldsymbol{r}) = \frac{\sum_{a\in C\cap A} x_{a,N}\cdot \boldsymbol{r}(a)}{\sum_{a\in C\cap A} x_{a,N}} \mid \Omega_C\right] = 1\,. \tag{5.8}$$

5.2. **y-values and transient behaviour.** 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:

Lemma 5.7. Let $\xi_N, N \subseteq [n]$ be strategies. Then every non-negative solution $y_a, y_{s,N}$, $a \in A, s \in S, N \subseteq [n]$ to Equation 1 effectively induces a strategy σ such that

$$\mathbb{P}^{\sigma}[\text{switch to } \xi_N \text{ in } s] = y_{s,N}$$

and σ is memoryless before the switch.

Proof. The idea is similar to [BBC+14, 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 $p \cdot \frac{y_{s,N}}{\sum_{N\subseteq[n]} y_{s,N}}$.

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

Then the strategy $\overline{\sigma}$ 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 $s_1, \ldots, s_k$. At this point, the strategy $\overline{\sigma}$ changes its behaviour as follows: First, it strives to reach s_1 with probability one. Upon reaching s_1, it chooses randomly with probability $\frac{y_{s_1,N}}{y_C}$ to behave as ξ_N forever, or otherwise to follow on to s_2. If the strategy $\overline{\sigma}$ chooses to go on to s_2, it strives to reach s_2 with probability one. Upon reaching s_2, it chooses with probability $\frac{y_{s_2,N}}{y_C - \sum_{N\subseteq[n]} y_{s_1,N}}$ 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

$$\frac{y_{s_i,N}}{y_C - \sum_{j=1}^{i-1}\sum_{N\subseteq[n]} y_{s_j,N}}\,.$$

Since ϑ stays in a MEC C with probability y_C, the probability that the strategy $\overline{\sigma}$ switches to ξ_N in s_i is equal to $y_{s_i,N}$. Further, as in [BBC+14] we can transform the part of $\overline{\sigma}$ before switching to ξ_N to a memoryless strategy and thus get strategy σ. □

Corollary 5.8. 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.

Proof. By Lemma 5.7 and Equation 3. □

5.3. **Proof of witnessing.** We now prove that the strategy σ of Corollary 5.8 with $\xi_N, N \subseteq [n]$ of Corollary 5.5 is indeed a witness strategy. Note that existence of ξ_N's depends on the sums of x-values being positive. This follows by Equation 2 and 3. We evaluate the strategy σ as follows:

$$\begin{aligned}
&\mathbb{E}^{\sigma}[\mathrm{lr}_{\inf}(\boldsymbol{r})] \\
&= \sum_{C \in \mathsf{MEC}} \sum_{N \subseteq [n]} \mathbb{P}^{\sigma}[\text{switch to } \xi_N \text{ in } C] \cdot \mathbb{E}^{\xi_N}[\mathrm{lr}_{\inf}(\boldsymbol{r}) \mid \Omega_C] \\
&\qquad\qquad \text{(by Equation 2, } \sum_{N \subseteq [n]} \mathbb{P}^{\sigma}[\text{switch to } \xi_N] = 1) \\
&= \sum_{C \in \mathsf{MEC}} \sum_{N \subseteq [n]} \Big(\sum_{a \in C \cap A} x_{a,N} \Big) \cdot \mathbb{E}^{\xi_N}[\mathrm{lr}_{\inf}(\boldsymbol{r}) \mid \Omega_C] \qquad \text{(by Corollary 5.8)} \\
&= \sum_{C \in \mathsf{MEC}} \sum_{\substack{N \subseteq [n]: \\ \sum_{a \in C \cap A} x_{a,N} > 0}} \Big(\sum_{a \in C \cap A} x_{a,N} \Big) \cdot \Big(\sum_{a \in C \cap A} x_{a,N} \cdot \boldsymbol{r}(a) / \sum_{a \in C \cap A} x_{a,N} \Big) \qquad \text{(by (5.7))} \\
&= \sum_{N \subseteq [n]} \sum_{C \in \mathsf{MEC}} \sum_{a \in C \cap A} x_{a,N} \cdot \boldsymbol{r}(a) \\
&= \sum_{N \subseteq [n]} \sum_{a \in A \cap \bigcup \mathsf{MEC}} x_{a,N} \cdot \boldsymbol{r}(a) \\
&= \sum_{N \subseteq [n]} \sum_{a \in A} x_{a,N} \cdot \boldsymbol{r}(a) \qquad \text{(by Lemma 5.1)} \\
&\geq \boldsymbol{exp} \qquad \text{(by Equation 5)}
\end{aligned}$$

and for each $i \in [n]$ we have

$$\begin{aligned}
&\mathbb{P}^{\sigma}[\mathrm{lr}_{\inf}(\boldsymbol{r})_i \geq \boldsymbol{sat}_i] = \\
&\sum_{C \in \mathsf{MEC}} \sum_{N \subseteq [n]} \mathbb{P}^{\sigma}[\text{switch to } \xi_N \text{ in } C] \cdot \mathbb{P}^{\xi_N}[\mathrm{lr}_{\inf}(\boldsymbol{r})_i \geq \boldsymbol{sat}_i \mid \Omega_C] \\
&\qquad\qquad \text{(by Equation 2, } \sum_{N \subseteq [n]} \mathbb{P}^{\sigma}[\text{switch to } \xi_N] = 1) \\
&= \sum_{C \in \mathsf{MEC}} \sum_{N \subseteq [n]} \Big(\sum_{a \in C \cap A} x_{a,N} \Big) \cdot \mathbb{P}^{\xi_N}[\mathrm{lr}_{\inf}(\boldsymbol{r})_i \geq \boldsymbol{sat}_i \mid \Omega_C] \qquad \text{(by Corollary 5.8)} \\
&= \sum_{C \in \mathsf{MEC}} \sum_{\substack{N \subseteq [n]: \\ \sum_{a \in C \cap A} x_{a,N} > 0}} \Big(\sum_{a \in C \cap A} x_{a,N} \Big) \cdot \mathbb{P}^{\xi_N} \Big[\sum_{a \in C \cap A} x_{a,N} \cdot \boldsymbol{r}(a)_i / \sum_{a \in C \cap A} x_{a,N} \geq \boldsymbol{sat}_i \Big] \\
&\qquad\qquad \text{(by (5.8))} \\
&\geq \sum_{C \in \mathsf{MEC}} \sum_{\substack{i \in N \subseteq [n]: \\ \sum_{a \in C \cap A} x_{a,N} > 0}} \Big(\sum_{a \in C \cap A} x_{a,N} \Big) \cdot \mathbb{P}^{\xi_N} \Big[\sum_{a \in C \cap A} x_{a,N} \cdot \boldsymbol{sat}_i / \sum_{a \in C \cap A} x_{a,N} \geq \boldsymbol{sat}_i \Big] \\
&\qquad\qquad \text{(by Equation 6)}
\end{aligned}$$

$$= \sum_{i \in N \subseteq [n]} \sum_{C \in \mathsf{MEC}} \sum_{a \in C \cap A} x_{a,N}$$
$$= \sum_{i \in N \subseteq [n]} \sum_{a \in A \cap \bigcup \mathsf{MEC}} x_{a,N}$$
$$= \sum_{i \in N \subseteq [n]} \sum_{a \in A} x_{a,N} \qquad \text{(by Lemma 5.1)}$$
$$\geq \boldsymbol{pr}_i \qquad \text{(by Equation 7)}$$

Remark 5.9. The proof of the corresponding claim for ε-witness strategies proceeds as above. We get that the strategy σ of Corollary 5.8 with $\zeta^{\varepsilon}_N, N \subseteq [n]$ of Lemma 5.3 is an ε-witness strategy. △

6. Algorithmic complexity

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

6.1. **Solution to (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 3.1, the upper bound on the algorithmic time complexity is polynomial in the number of variables in system L. Hence, the realizability problem for **(multi-quant-conjunctive)** can be decided in time polynomial in $|G|$ and exponential in n.

6.2. **Solution to (multi-quant-joint) and the special cases.** In order to decide **(multi-quant-joint)**, 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 4.1). 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.

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

(1) transient flow: for $s \in S$
$$\mathbb{1}_{s_0}(s) + \sum_{a \in A} y_a \cdot \delta(a)(s) = \sum_{a \in Act(s)} y_a + y_{s,\emptyset} + y_{s,[n]}$$
(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:

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

Since there are now $\mathcal{O}(|G| \cdot n)$ variables, the problem as well as its special cases can be decided in polynomial time.

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.

Theorem 6.1. The **(multi-quant-joint)** realizability problem (and thus also all its special cases) can be decided in time polynomial in $|G|$ and n. □

6.3. **Solution to (multi-quant-conjunctive-joint).** 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, the program is the following:

(1) transient flow: for $s \in S$

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

(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:

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

(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 $i \in [n]$

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

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, respectively):

($\widetilde{6}$)

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

($\widetilde{7}$)

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

Note that the number of variables is double that for **(multi-quant-conjunctive)**. Therefore, the complexity remains essentially the same:

Corollary 6.2. The algorithmic complexity for the **(multi-quant-conjuctive-joint)** is polynomial in the size of the MDP and exponential in n. □

Remark 6.3. The strategies for the case of **(multi-quant-conjunctive-joint)** are very similar to that of **(multi-quant-conjunctive)**. Indeed, the structure of the constructed (ε-)witness strategies is the same: the memoryless strategy for reaching the desired MECs is followed by a stochastic-update switch to strategies for the recurrent behaviour. The only difference is the following. (ε-)witness strategies for **(multi-quant-conjunctive)** switch to strategies ξ_N (or ζ^{ε}_N), each given by values of x-variables indexed by a fixed $N \subseteq [n]$. In contrast, strategies for **(multi-quant-conjunctive-joint)** switch to strategies $\xi_{N,b}$ (or $\zeta^{\varepsilon}_{N,b}$), each given by values of x-variables indexed by a fixed $N \subseteq [n]$ *and* $b \in \{yes, no\}$. $\triangle$

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 number of variables is then exponential in the number of constraints rather than just in the dimension of the rewards.

6.4. **Hardness.** The **(multi-quant-conjunctive-joint)** problem is also of significant theoretical interest since we can also prove the following hardness result:

Theorem 6.4. The **(multi-quant-conjunctive-joint)** problem 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 $C = \{c_1, \ldots, c_k\}$ over atomic propositions $Ap = \{a_1, \ldots, a_p\}$. We denote $\overline{Ap} = \{\overline{a_1}, \ldots, \overline{a_p}\}$ the literals that are negations of the atomic propositions.

We define an MDP $G_\varphi = (S, A, Act, \delta, s_0)$ as follows:

- $S = \{s_i \mid i \in [p]\}$,
- $A = Ap \cup \overline{Ap}$,
- $Act(s_i) = \{a_i, \overline{a_i}\}$ for $i \in [p]$,
- $\delta(a_i)(s_{i+1}) = 1$ and $\delta(\overline{a_i})(s_{i+1}) = 1$ (actions are assigned Dirac distributions),
- $s_0 = s_1 = s_{p+1}$.

The constructed MDP is illustrated in Fig. 6. Intuitively, a run in G_φ repetitively chooses a valuation.

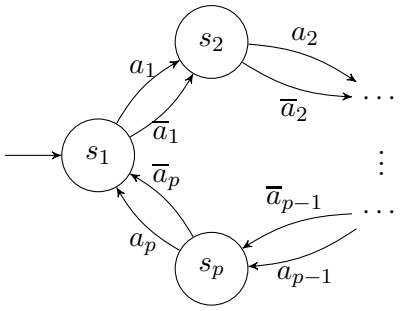

FIGURE 6. MDP G_φ

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:

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

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 simply count the number of uses of a literal; thus $\mathrm{lr}_{\inf}(\boldsymbol{r})_a = \boldsymbol{Freq}_a$.

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

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

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

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 $\overline{a_i}$ happens every p steps, runs that mix both with positive frequency cannot exceed the value threshold $1/p$. Therefore, half of the runs must use only a_i, half must use only $\overline{a_i}$. 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 $\overline{\overline{a}}$ is identified with a).

Therefore, each literal has frequency $1/p$ 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

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

which follows from the following lemma:

Lemma 6.5. For every witness strategy σ satisfying the (conjunctive-S) constraint, and for each $a \in Ap$, we have

$$\mathbb{P}^\sigma\left[\boldsymbol{Freq}_a = \frac{1}{p} \text{ and } \boldsymbol{Freq}_{\overline{a}} = 0\right] + \mathbb{P}^\sigma\left[\boldsymbol{Freq}_a = 0 \text{ and } \boldsymbol{Freq}_{\overline{a}} = \frac{1}{p}\right] = 1\,.$$

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

$$\boldsymbol{Freq}_a + \boldsymbol{Freq}_{\overline{a}} \leq 1/p \tag{6.1}$$

for every run. Indeed, $\boldsymbol{Freq}_a + \boldsymbol{Freq}_{\overline{a}} = \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_a + \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_{\overline{a}} \leq \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}(\mathbb{1}_a + \mathbb{1}_{\overline{a}}) = 1/p$.

Therefore, the two events $\boldsymbol{Freq}_a \geq 1/p$ and $\boldsymbol{Freq}_{\overline{a}} \geq 1/p$ are disjoint. Due to the (conjunctive-S) constraint, almost surely exactly one of the events occurs. Indeed,

$$1 \geq \mathbb{P}^\sigma\left[\boldsymbol{Freq}_a \geq \frac{1}{p} \cup \boldsymbol{Freq}_{\overline{a}} \geq \frac{1}{p}\right] = \mathbb{P}^\sigma\left[\boldsymbol{Freq}_a \geq \frac{1}{p}\right] + \mathbb{P}^\sigma\left[\boldsymbol{Freq}_{\overline{a}} \geq \frac{1}{p}\right] \geq \frac{1}{2} + \frac{1}{2} = 1$$

with the equality by disjointness of the events and the last inequality by (conjunctive-S).

Therefore, by (6.1), almost surely either $\boldsymbol{Freq}_a = 1/p$ and $\boldsymbol{Freq}_{\overline{a}} = 0$, or $\boldsymbol{Freq}_a = 0$ and $\boldsymbol{Freq}_{\overline{a}} = 1/p$. □

By the (joint-S) constraint, we have a set Ω_{sat}, with non-zero measure, of runs satisfying $\mathrm{lr}_{\inf}(\boldsymbol{r})_c \geq 1$ 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 $\mathrm{lr}_{\inf}(\boldsymbol{r})(\omega)_c \geq 1$, there is an action ℓ such that

- $\boldsymbol{Freq}_\ell(\omega) > 0$, 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. □

Theorem 6.4 contrasts Theorem 6.1: 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 3.1 (though our lower bound does not work for (conjunctive-SAT) with (EXP)).

The results are summarized in Table 2 and contrasted to the previously known polynomial bounds in Table 1.

7. Strategy complexity

First, we recall the structure of witness strategies generated from L in Section 5. 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.

7.1. **Randomization and memory.** Similarly to the traditional setting with the expectation or the satisfaction semantics considered separately, the case with a single objective is simpler.

Lemma 7.1. 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 single long-run expected average objective, and thus also to be deterministic and memoryless [Put94]. Intuitively, in this case each MEC is either stayed at almost surely, or left almost surely if the value of the outgoing action is higher. □

Further, both for the expectation and the satisfaction semantics, deterministic memoryless strategies are sufficient for *quantitative* queries [FV97, BBE10] with single objective. In contrast, we show that both randomization and memory is necessary in our combined setting even for ε-witness strategies.

Example 7.2. 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. 7. 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.

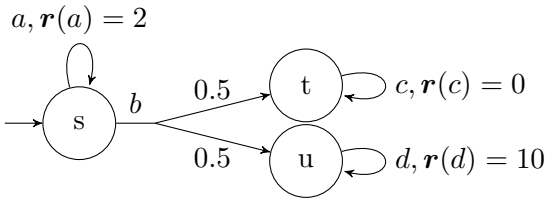

FIGURE 7. An MDP with a single objective, where both randomization and memory is necessary △

In the rest of the section, we discuss bounds on the size of the memory and the degree of randomization. Due to [BBC+14, Section 5], infinite memory is indeed necessary for witnessing (joint-SAT) with $pr = 1$, hence also for **(multi-qual)**.

7.2. **Memory bounds for deterministic update.** We prove that finite memory is sufficient in several cases, namely for all ε-witness strategies and for **(mono-quant)** witness strategies. Moreover, these results also hold for deterministic-update strategies. Indeed, as one of our technical contributions, we prove that stochastic update at the moment of switching is not necessary and deterministic update is sufficient, requiring only a finite blow up in the memory size.

Lemma 7.3. 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 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 $rate = \sum_{s \notin C} \delta(a)(s)$ be the probability to actually leave C when taking a once. Then to achieve the overall

probability *leave* of leaving we can reach s with $a \in Act(s)$ and play a with probability 1 and repeat this m times for some $m \in \mathbb{N}$ (if $leave = 1$ then $m = \infty$) and finally reach s once more and play a with probability $p \in [0,1]$ and an action staying in C with the remaining probability. We now define m and p. If $rate = 1$ then $m = 0$ and $p = leave$. Assume $rate < 1$. Then we must ensure that the probability not to leave via a be

$$1 - leave = (1 - rate)^m \cdot \big(p(1 - rate) + (1 - p)\big) \tag{7.1}$$

Indeed, $(1 - rate)^m$ stands for failing to leave m-times, and the last time we either choose a and fail again or not choose a at all. This requirement is equivalent to

$$m = \frac{\ln(1 - leave) - \ln(1 - p \cdot rate)}{\ln(1 - rate)}$$

For $p \in [0,1]$ we have also $\frac{\ln(1-p\cdot rate)}{\ln(1-rate)} \in [0,1]$. Therfore, in order to choose $m \in \mathbb{N}$, we can simply set $m := \lfloor \frac{\ln(1-leave)}{\ln(1-rate)} \rfloor$, which also ensures that $p \in [0,1]$ for the respective $p := \frac{1}{rate}(1 - \frac{1-leave}{(1-rate)^m})$, obtained from (7.1).

In order to implement the strategy in MECs of this second type, for each action it is sufficient to have a counter up to the respective m. □

Remark 7.4. Moreover, our proof also shows, that finite memory is sufficient before switching to ξ_N's (as defined in Section 5) for deterministic-update witnessing (and ε-witnessing) strategies. Therefore, finite memory deterministic update is sufficient for $\varepsilon-$witness strategies, in particular also for (joint-SAT), which improves the strategy complexity known from [BBC+14]. Note that in general, conversion of a stochastic-update strategy to a deterministic-update strategy requires an infinite blow up in the memory [dAHK07]. △

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

Lemma 7.5. 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 $\zeta_N^{1/2^i}$. □

Remark 7.6. The previous proof of sufficiency of deterministic-update finite memory for ε-witness strategies applies also to **(multi-quant-conjunctive-joint)**. Indeed, firstly, Lemma 7.3 applies verbatim to **(multi-quant-conjunctive-joint)**. Secondly, we switch to only finitely many recurrent strategies due to Remark 6.3. △

Secondly, infinite memory is required only for multiple objectives:

Lemma 7.7. 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-objective case.

Example 7.8. For deterministic-update ε-witness strategies for **(mono-quant)** problem, memory with size dependent on the transition probabilities is necessary. Indeed, consider the same realizability problem as in Example 7.2, but with a slightly modified MDP parametrized by λ, depicted in Fig. 8. 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 [BBC$^+$14], e.g. $\boldsymbol{exp} = (0.5, 0.5)$ and the MDP of Fig. 9. Therefore, we provide a lower bound also for this simpler case (no MDP-dependent lower bound is provided in [BBC$^+$14]).

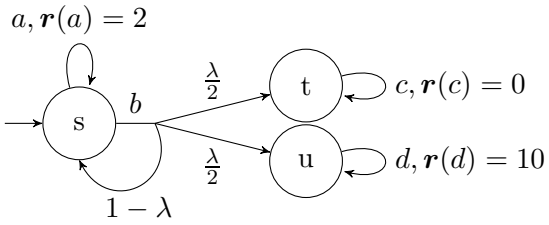

FIGURE 8. An MDP family with a single objective, where memory with size dependent on transition probabilities is necessary for deterministic-update strategies

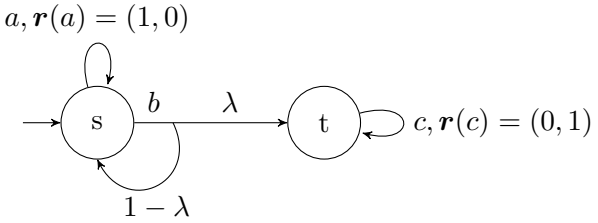

FIGURE 9. An MDP family, where memory with size dependent on transition probabilities is necessary for deterministic-update strategies even for (EXP) studied in [BBC$^+$14]

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 $p > 0$, 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. △

7.3. **Memory bounds for stochastic update.** Although we have shown that stochastic update is not necessary, it may be helpful when memory is small.

Lemma 7.9. Stochastic-update 2-memory strategies are sufficient for witness strategies for **(mono-quant)**.

Proof. The strategy σ of Section 5, which reaches the MECs and stays in them with given probability, is memoryless up to the point of switch by Corollary 5.8. Further, we can achieve the optimal value in each MEC using a memoryless strategy as in Lemma 7.7. □

Theorem 7.10. Upper bounds on memory size for stochastic-update ε-witness strategies are as follows:

- **(multi-qual)** 2 memory elements,
- **(multi-quant-joint)** 3 memory elements,
- **(multi-quant-conjunctive)** $2^n + 1$ memory elements,
- **(multi-quant-conjunctive-joint)** $2^{n+1} + 1$ memory elements.

Proof. The structure of ε-witness strategies is described in Remark 5.9. Let us recall from Corollary 5.8 that strategy σ is memoryless before the switch. For **(multi-qual)**, **(multi-quant-joint)** and **(multi-quant-conjunctive)**, we perform the stochastic-update switch to different memory elements corresponding to the different strategies ζ^{ε}_{N}. From Lemma 5.3 we have that every such strategy ζ^{ε}_{N} is also memoryless. From Lemma 5.7 we have that we switch only to such ζ^{ε}_{N} for $N \subseteq [n]$, which correspond to possible nonzero variables $y_{s,N}$. Therefore, the number of memory elements needed is the number of possible nonzero variables $y_{s,N}$ for $N \subseteq [n]$ and additionally one element for the strategy σ before the switch.

Altogether, we get the following upper bounds on memory size of ε-witness strategies. For **(multi-quant-conjunctive)**, 2^n+1 memory elements are sufficient, since all of the $y_{s,N}$ for $N \subseteq [n]$ can be positive. For **(multi-quant-joint)**, 3 memory elements are sufficient, because we use only $y_{s,[n]}$ and $y_{s,\emptyset}$ as discussed in 6.2. Finally for **(multi-qual)**, 2 memory elements are sufficient, because we use only y_s as in 3.2.1.

Due to Remark 6.3, the bound on the number of recurrent strategies for **(multi-quant-conjunctive-joint)** is twice as large as for **(multi-quant-conjunctive)**, i.e., 2^{n+1}. The upper bound on the size of memory for ε-witness strategies for **(multi-quant-conjunctive-joint)** is thus $1 + 2^{n+1}$, compared to $1 + 2^n$ for **(multi-quant-conjunctive)**. □

Example 7.11. For **(multi-quant-joint)**, ε-witness strategies may require memory with at least 3 elements. Consider an MDP with two states s and t with transitions and rewards as depicted in Fig. 10. Further, let $\boldsymbol{sat} = (1, 0, 0)$, $\boldsymbol{pr} = \frac{1}{2}$ and $\boldsymbol{exp} = (0, 1, 1)$.

Suppose 2 memory elements are sufficient. In state s for each memory element we can either stay in s or go with some positive probability to state t. Therefore we have three cases on the behaviour in s regarding the transition to t:

(1) for each memory element we have positive probability p_1 and p_2 respectively, to go to state t,
(2) for both memory elements we have zero probability to go to t and
(3) for one memory element, say memory element 1, we have zero probability and for the other one, say memory element 2, we have positive probability p to go to t.

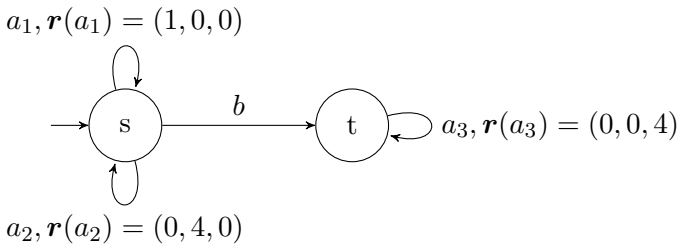

FIGURE 10. An MDP where 3-memory is necessary for **(multi-quant-joint)**

In the first case, we go to t eventually almost surely. Indeed, in each step we enter t with probability at least $\min(p_1, p_2)$ and cannot return back. Therefore, we stay in t forever and thus we cannot satisfy the satisfaction constraint.

In the second case, we never enter state t. Hence, we cannot satisfy the expectation constraint, because $\boldsymbol{r}(a_1)_3 = \boldsymbol{r}(a_2)_3 = 0$.

In the third case, we firstly assume that we switch from memory 1 to 2 with some positive probability p_1. Then in each step we have at least probability $p_1 \cdot p$ to enter t. Therefore, we end up in state t almost surely, not satisfying constraints, as shown above. Secondly, suppose we cannot switch from memory 1 to 2. Then we almost surely end up in state s with memory 1 or in state t. In state s with memory 1 we can either play action a_1 with probability 1 or with smaller potentially zero probability q. In the former case, $\mathrm{lr}(\boldsymbol{r}_2) = 0$, thus violating the expectation constraint. In the latter case, for almost every run $\mathrm{lr}(\boldsymbol{r}_1) \leq 1 - q$, contradicting the satisfaction constraint.

Note that a witnessing strategy exists, which uses only 3 memory elements. On half of the runs, we play only action a_1 to satisfy the satisfaction constraint. So we define $\sigma_n(s,1)(a_1) = 1$. To satisfy the expectation constraint for $\boldsymbol{r}_2$ we define $\sigma_n(s,2)(a_2) = 1$. With the last memory element we want to satisfy the expectation constraint for $\boldsymbol{r}_3$ and thus we define $\sigma_n(s,3)(b) = 1$ and $\sigma_n(t,3)(a_3) = 1$. We define the initial distribution by $\alpha(1) = \frac{1}{2}$, $\alpha(2) = \frac{1}{4}$ and $\alpha(3) = \frac{1}{4}$ and therefore the memory update function not to change memory. Consequently, the achieved expectation is $(\frac{1}{2} \cdot 1, \frac{1}{4} \cdot 4, \frac{1}{4} \cdot 4) \geq \boldsymbol{exp}$. △

However, even with stochastic update, the size of the finite memory cannot be bounded by a constant for **(multi-quant-conjunctive)**.

Example 7.12. Even ε-witness strategy for **(multi-quant-conjunctive)** 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. 11 illustrates the case with $n = 3$. Further, let $\boldsymbol{sat} = \mathbf{1}$ and $\boldsymbol{pr} = 1/n \cdot \mathbf{1}$.

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 for witness and ε-witness strategies in Table 3 and Table 4, respectively.

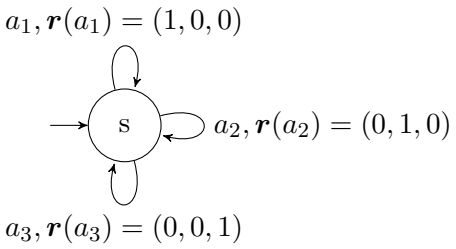

FIGURE 11. An MDP where n-memory is necessary, depicted for $n = 3$

8. Pareto curve approximation and complexity summary

For a single objective, 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 objectives we obtain the following:

Theorem 8.1. For $\varepsilon > 0$, an ε-approximation of the Pareto curve for **(multi-quant-conjunctive-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 [PY00], 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 $\boldsymbol{v}$). □

The proof of Theorem 8.1 shows that we can obtain a Pareto-curve approximation also 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-objective case, we can compute also the optimal $\boldsymbol{pr}$ given $\boldsymbol{exp}$ and $\boldsymbol{sat}$, or the optimal $\boldsymbol{sat}$ given $\boldsymbol{pr}$ and $\boldsymbol{exp}$.

The complexity results are summarized in the following theorem:

Theorem 8.2. The algorithmic complexities are shown in Table 2. The bounds on the complexity of the witness and ε-witness strategies are as shown in Table 3 and Table 4, respectively.

Comments on the tables. **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 or generalization relation depicted in Fig. 1 and for Table 3 and 4 by $\varepsilon-$witness strategies being a weaker notion than witness strategies. The abbreviations stoch.-up., det.-up., rand., det., inf., fin., and X-mem. stand for stochastic update, deterministic update, randomizing, deterministic, infinite-, finite- and X-memory strategies, respectively. Here n is the dimension of reward function and $p = 1/p_{\min}$ where $p_{\min}$ is the smallest positive probability in the MDP. Note that inf. actually means that the strategy is in form of a Markov strategy, see Section 5.

Remark 8.3. For a comparison, the results on previously studied subcases of our problems are depicted in Table 1. △

TABLE 1. Previous results on algorithmic and strategy complexities. The abbreviations alg., strat., and c. stand for algorithmic, strategy, and complexity, respectively. Cases multiple and single refer to the number of objectives. Results for single-objective MDPs are based on classical literature, e.g. [Put94, Thm.9.1.8]. Results for MDPs with multiple objectives are due to [BBC+14].

Case	Alg. c.	Witness strat. c.	ε-witness strat. c.
multiple (joint-SAT)	$poly(\|G\|, n)$	**U:** det.-up. inf. **L:** rand. inf.	**U:** stoch.-up. 2-mem. **L:** rand. 2-mem.
multiple (EXP)	$poly(\|G\|, n)$	**U:** det.-up. inf. **L:** rand. inf.	**U:** stoch.-up. 2-mem., det.-up. fin. **L:** rand. 2-mem.
single (joint-SAT)	$poly(\|G\|)$	**U=L:** det. 1-mem.	**U=L:** det. 1-mem.
single (EXP)	$poly(\|G\|)$	**U=L:** det. 1-mem.	**U=L:** det. 1-mem.

TABLE 2. Algorithmic complexity results for each of the discussed cases.

Case	Algorithmic complexity
(multi-quant-conj.-joint)	$poly(\|G\|, 2^n)$ [Cor.6.2], NP-hard [Thm. 6.4]
(multi-quant-conj.)	$poly(\|G\|, 2^n)$ [Thm.3.1]
(multi-quant-joint)	$poly(\|G\|, n)$ [Thm.6.1]
(multi-qual)	$poly(\|G\|, n)$
(mono-quant)	$poly(\|G\|)$
(mono-qual)	$poly(\|G\|)$

9. CONCLUSION

We have presented a unifying solution framework to the expectation and satisfaction optimization of Markov decision processes with multiple objectives. This allows us to synthesize optimal and ε-optimal risk-averse strategies. We have considered several possible combinations of the two semantics and provided algorithms for their solution as well as the complete picture of the complexities for all these cases.

Regarding the algorithmic complexity, we have shown that **(multi-quant-joint)** and all its special cases can be solved in polynomial time. For both **(multi-quant-conjunctive)** and **(multi-quant-conjunctive-joint)**, we have presented an algorithm that works in time polynomial in the size of MDP, but exponential in the dimension of reward function. However, the exponential in the dimension of reward function is not a limitation for most of practical purposes since the dimension is typically low. For the latter case we have also proved that the problem is NP-hard. The complexity of **(multi-quant-conjunctive)** remains an

TABLE 3. Witness strategy complexity bounds for each of the discussed cases.

Case	Witness strategy complexity
(multi-quant-conj.-joint)	**U:** det.-up. [Rem.7.6] inf. **L:** rand. inf.
(multi-quant-conj.)	**U:** det.-up. [Lem.7.3] inf. **L:** rand. inf.
(multi-quant-joint)	**U:** det.-up. inf. **L:** rand. inf.
(multi-qual)	**U:** det.-up. inf. **L:** rand. inf. [BBC$^+$14, Sec.5]
(mono-quant)	**U:** stoch.-up. 2-mem. [Lem.7.9], det.-up. fin. [Lem.7.7] **L:** rand. 2-mem., for det.-up. p-mem.
(mono-qual)	**U:** (trivially also **L:**) det. 1-mem. [Lem.7.1]

TABLE 4. ε-witness strategy complexity bounds for each of the discussed cases.

Case	ε-witness strategy complexity
(multi-quant-conj.-joint)	**U:** stoch.-up. $(2^{n+1}+1)$-mem. [Thm.7.10], det.-up. fin. [Rem.7.6] **L:** rand. n-mem. [Ex.7.12], for det.-up. p-mem.
(multi-quant-conj.)	**U:** stoch.-up. (2^n+1)-mem. [Thm.7.10], det.-up. fin. [Lem.7.5] **L:** rand. n-mem. [Ex.7.12], for det.-up. p-mem.
(multi-quant-joint)	**U:** stoch.-up. 3-mem. [Thm.7.10], det.-up. fin. **L:** rand. 3-mem. [Ex.7.11]
(multi-qual)	**U:** stoch.-up. 2-mem. [Thm.7.10], det.-up. fin. **L:** rand. mem. [BBC$^+$14, Sec.3]
(mono-quant)	**U:** stoch.-up. 2-mem., det.-up. fin. **L:** rand. [Ex.7.2] 2-mem. [Ex.7.2], for det.-up. p-mem. [Ex.7.8]
(mono-qual)	**U:** (trivially also **L:**) det. 1-mem.

interesting open question. Moreover, our algorithms for Pareto-curve approximation work in time polynomial in the size of MDPs and exponential in the dimension of reward function. However, note that even for the special case of expectation semantics the current best known algorithms depend exponentially on the dimension of reward function [BBC$^+$14].

We have also provided comprehensive results on strategy complexities. It is known that for both expectation and satisfaction semantics with single objective, deterministic memoryless strategies are sufficient [FV97, BBE10, BBC$^+$14]. We have shown this carries over in the **(mono-qual)** case only. In contrast, for **(mono-quant)** both randomization and memory is necessary. However, we have also shown that only a restricted form of randomization (deterministic update) is necessary even for **(multi-quant)**, thus improving the upper bound for $\varepsilon-$witness strategies for the satisfaction problem of [BBC$^+$14] to finite-memory deterministic update. Furthemore, we have established that with deterministic update the memory size is dependent on the MDP; the result also applies to the expectation problem of [BBC$^+$14], where no MDP-dependent lower bound was given. We have presented upper bounds on stochastic update $\varepsilon-$witness strategies, which are constant for **(multi-qual)** and **(multi-quant-joint)**, and exponentially dependent on the dimension of reward

function for **(multi-quant-conjunctive)** and **(multi-quant-conjunctive-joint)**. The question whether there are polynomially dependent upper bounds for the latter two cases stays open.

Acknowledgements We are very thankful to the anonymous reviewers for their helpful suggestions and pointing at gaps in the proofs of Lemma 4.3 and Lemma 5.1, and to Rasmus Ibsen-Jensen for discussing the proof of Lemma 4.2.

References

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

[BBC+14] T. Brázdil, V. Brožek, K. Chatterjee, V. Forejt, and A. Kučera. Markov decision processes with multiple long-run average objectives. *LMCS*, 10(1), 2014.

[BBE10] T. Brázdil, V. Brožek, and K. Etessami. One-counter stochastic games. In *FSTTCS*, pages 108–119, 2010.

[BCFK13] T. Brázdil, K. Chatterjee, V. Forejt, and A. Kučera. Trading performance for stability in Markov decision processes. In *LICS*, pages 331–340, 2013.

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

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

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

[CH11] K. Chatterjee and M. Henzinger. Faster and dynamic algorithms for maximal end-component decomposition and related graph problems in probabilistic verification. In *SODA*, pages 1318–1336, 2011.

[CH12] K. Chatterjee and M. Henzinger. An $O(n^2)$ time algorithm for alternating Büchi games. In *SODA*, pages 1386–1399, 2012.

[CH14] K. Chatterjee and M. Henzinger. Efficient and dynamic algorithms for alternating Büchi games and maximal end-component decomposition. *JACM*, 2014.

[Cha07] K. Chatterjee. Markov decision processes with multiple long-run average objectives. In *FSTTCS*, pages 473–484, 2007.

[CL13] K. Chatterjee and J. Lacki. Faster algorithms for Markov decision processes with low treewidth. In *CAV*, pages 543–558, 2013.

[CMH06] K. Chatterjee, R. Majumdar, and T. A. Henzinger. Markov decision processes with multiple objectives. In *STACS*, pages 325–336, 2006.

[CR15] Lorenzo Clemente and Jean-François Raskin. Multidimensional beyond worst-case and almost-sure problems for mean-payoff objectives. In *LICS*, pages 257–268, 2015.

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

[CY98] C. Courcoubetis and M. Yannakakis. Markov decision processes and regular events. *Automatic Control, IEEE Transactions on*, 43(10):1399–1418, October 1998.

[dA97] L. de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, 1997.

[dAHK07] L. de Alfaro, T. A. Henzinger, and O. Kupferman. Concurrent reachability games. *Theor. Comput. Sci*, 386(3):188–217, 2007.

[DEKM98] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge Univ. Press, 1998.

[EKVY08] K. Etessami, M. Kwiatkowska, M. Vardi, and M. Yannakakis. Multi-objective model checking of Markov decision processes. *LMCS*, 4(4):1–21, 2008.

[FKN+11] V. Forejt, M. Z. Kwiatkowska, G. Norman, D. Parker, and H. Qu. Quantitative multi-objective verification for probabilistic systems. In *TACAS*, pages 112–127, 2011.

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

[FKR95] J. A. Filar, D. Krass, and K. W Ross. Percentile performance criteria for limiting average Markov decision processes. *Automatic Control, IEEE Transactions on*, 40(1):2–10, Jan 1995.

[FV97] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, 1997.
[How60] H. Howard. *Dynamic Programming and Markov Processes*. MIT Press, 1960.
[KGFP09] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas. Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics*, 25(6):1370–1381, 2009.
[KNP02] M. Kwiatkowska, G. Norman, and D. Parker. PRISM: Probabilistic symbolic model checker. In *TOOLS' 02*, pages 200–204, 2002.
[Kos88] J. Koski. Multicriteria truss optimization. In *Multicriteria Optimization in Engineering and in the Sciences*. 1988.
[Owe95] G. Owen. *Game Theory*. Academic Press, 1995.
[Put94] M.L. Puterman. *Markov Decision Processes*. John Wiley and Sons, 1994.
[PY00] C. H. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *FOCS*, pages 86–92, 2000.
[Roy88] H. Royden. *Real Analysis*. Prentice Hall, 3rd edition, 12 February 1988.
[RRS15] Mickael Randour, Jean-François Raskin, and Ocan Sankur. Percentile queries in multi-dimensional markov decision processes. In *CAV, Part I*, pages 123–139, 2015.
[Sch86] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley-Interscience, 1986.
[SCK04] R. Szymanek, F. Catthoor, and K. Kuchcinski. Time-energy design space exploration for multi-layer memory architectures. In *DATE*, pages 318–323, 2004.
[Seg95] R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, MIT, 1995.
[Var85] M. Vardi. Automatic verification of probabilistic concurrent finite state programs. In *FOCS*, pages 327–338, 1985.
[WL99] C. Wu and Y. Lin. Minimizing risk models in Markov decision processes with policies depending on target values. *Journal of Mathematical Analysis and Applications*, 231(1):47–67, 1999.
[YC03] P. Yang and F. Catthoor. Pareto-optimization-based run-time task scheduling for embedded systems. In *CODES+ISSS*, pages 120–125, 2003.

Appendix A. Limear program for the running example

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

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