

Stochastic Games with Disjunctions of Multiple Objectives*

Tobias Winkler

RWTH Aachen University, Germany
tobias.winkler@cs.rwth-aachen.de

Maximilian Weininger

Technical University of Munich, Germany
maxi.weininger@tum.de

Stochastic games combine controllable and adversarial non-determinism with stochastic behavior and are a common tool in control, verification and synthesis of reactive systems facing uncertainty. Multi-objective stochastic games are natural in situations where several—possibly conflicting—performance criteria like time and energy consumption are relevant. Such *conjunctive* combinations are the most studied multi-objective setting in the literature. In this paper, we consider the dual *disjunctive* problem. More concretely, we study turn-based stochastic two-player games on graphs where the winning condition is to guarantee at least one reachability or safety objective from a given set of alternatives. We present a fine-grained overview of strategy and computational complexity of such *disjunctive queries* (DQs) and provide new lower and upper bounds for several variants of the problem, significantly extending previous works. We also propose a novel value iteration-style algorithm for approximating the set of Pareto optimal thresholds for a given DQ.

1 Introduction

Stochastic games (SG), e.g. [21, 36], combine controllable and adversarial non-determinism with stochastic behavior. In their turn-based two-player version, SGs are played on graphs where the vertices are called states, and every state either belongs to one of the two players Eve and Adam, or is controlled by a probabilistic environment. In each round, the player in control of the state chooses an action—an edge of the graph—and the game transitions to the successor state. In probabilistic states, the successor is sampled according to a fixed observable distribution over the outgoing edges. *Simple* SGs [21] have just a single *reachability* objective. A key question is whether Eve can control her states such that a target is reached with at least a certain probability, no matter the behavior of the opponent Adam. Dually, Adam has a *safety objective* in this setting: He should maximize the chance of staying in the—from his point of view—safe region, i.e., avoiding Eve’s target states.

Multi-objective stochastic games [19] extend this by allowing Boolean combinations of several different probability or expectation thresholds on various objectives. Such games have been used to synthesize optimal controllers in application scenarios where the system at hand is exposed to an environment with both stochastic and non-deterministic aspects [6, 20, 26]. A natural subclass of multi-objectives are *disjunctive queries* (DQ) [19, 27] where the player has to satisfy at least one alternative from a given set of options.

In this paper, we study DQs with both reachability and safety options. More specifically, given a game and a finite set of reachability and safety objectives, each equipped with a desired threshold probability, we ask whether Eve can satisfy at least one option with probability at least the respective threshold. Our motivation for studying DQs is twofold. (1) DQs are interesting in their own right as they allow for a fine-grained specification of alternatives over desirable outcomes of a controlled stochastic system. (2) DQs are equivalent to the more widely used *conjunctive queries* (CQs) under an alternative

*This work was supported by the DFG RTG 2236 “UnRAVeL” (Winkler) and DFG projects 383882557 “SUV” and 427755713 “GOPro” (Weininger).

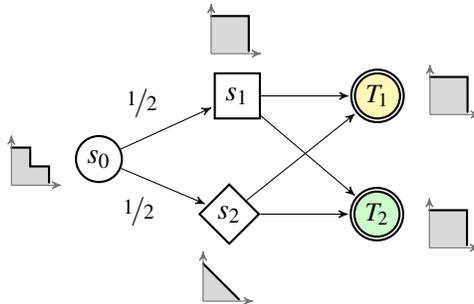


Figure 1: Example SG with a disjunctive query $\mathbb{P}(\diamond T_1) \geq x \vee \mathbb{P}(\diamond T_2) \geq y$. The Pareto sets, i.e., the feasible threshold vectors (x,y) are depicted next to the respective states.

semantics, namely if the opponent Adam has to *reveal* his strategy to Eve before the game starts. This amounts to changing the quantification order over strategies from $\exists\forall$ —which is the standard order—to $\forall\exists$. In general, this makes a difference since multi-objective SGs are not *determined* [19]. This optimistic “*asserted-exposure*” ($\forall\exists$) semantics is interesting in situations where the non-deterministic system state can actually be observed at any given point in time. For instance, in the smart heating case study from [34], the position of the doors in a house is non-deterministically controlled by Adam and not directly observable. However, if door sensors were to be installed, the $\forall\exists$ -semantics would be more adequate since the door positions are now observable (but still uncontrollable).

The technical intricacies of DQs are best illustrated by means of an example: Consider the SG in Figure 1. It comprises 5 states: the probabilistic state s_0 , states s_1 and s_2 controlled by Eve and Adam, respectively, and the two targets labeled T_1 and T_2 . Suppose that Eve’s objective is the DQ “reach T_1 with probability at least x or T_2 with probability at least y ”, in symbols $\mathbb{P}(\diamond T_1) \geq x \vee \mathbb{P}(\diamond T_2) \geq y$. The coordinate systems next to the states show their *Pareto sets*, i.e., the set of threshold vectors (x,y) for which Eve can win the DQ, assuming the game starts in that state. For s_1 , the Pareto set is the whole box $[0, 1]^2$ since Eve can reach either of the targets surely from s_1 by picking the respective action. For s_2 , the Pareto set contains all convex combinations of $(0, 1)$ and $(1, 0)$ and all point-wise smaller vectors, forming a triangle. This is because Adam has to distribute the whole probability mass *somewhere*; for any threshold vector (x,y) , $x + y \leq 1$, he cannot avoid satisfying it. However, if $x + y > 1$, Adam can prevent Eve from winning in state s_2 : For example, if $(x,y) = (0.6, 0.6)$, Adam can randomize equally between both actions, and no target is reached with at least 0.6. This also shows that solving a DQ is *not* equivalent to solving each objective separately: From s_2 , Eve can neither guarantee that T_1 nor T_2 is reached with positive probability; however, she can guarantee the thresholds $(0.5, 0.5)$ in the DQ. Finally, Eve can achieve each threshold vector (x,y) with $x \leq 0.5$ or $y \leq 0.5$ from s_0 because s_1 is reached with probability 0.5 where she can put all probability mass on one of the targets. On the other hand, Eve cannot ensure any vector (x,y) with $x > 0.5 \wedge y > 0.5$ because once she has fixed a strategy, Adam can mirror it so that both targets are reached with *exactly* 0.5.

The example in Figure 1 demonstrates two important properties of disjunctive queries in SGs: Firstly, Pareto frontiers are not necessarily convex, as in state s_0 . This is in contrast to *conjunctive queries* where the set of achievable probability thresholds is always convex [19, 24]. Secondly, as mentioned above, SGs with multiple objectives are in general not *determined* [19], i.e., it is relevant which player fixes their strategy first, thereby *revealing* it to the other player before the game starts. In fact, in the example above, switching the quantification order allows Eve to take advantage by reacting to the strategy of Adam. For

instance, she could then ensure that at least one of the targets is reached with probability at least 0.75.

Contributions and overview In summary, this paper makes the following contributions:

- A comprehensive overview of strategy (Section 3, Table 1) and computational complexity (Section 4, Table 2) of disjunctive reachability-safety queries in stochastic games, significantly extending previous results from the literature [19, 27, 35]. In particular, motivated by the observation that randomized strategies are undesirable or meaningless for certain applications (e.g., medical or product design [23]), we study the setting of DQs under *deterministic* strategies for both players. Notably, this lead to rather high complexities: Qualitative queries are PSPACE-hard and quantitative reachability is even undecidable.
- A value iteration-style algorithm in the vein of [19] for approximating the Pareto sets of DQs or CQs under the alternative asserted-exposure semantics (Section 5).

Related work SGs were introduced by Shapley [36] in 1953. Simple SGs—the turn-based variant with a reachability objective—are one of the intriguing problems in $NP \cap coNP$ but not known to be in P [22]. See [21, 30] for an overview of solution algorithms. Turn-based SGs were studied with a variety of other objectives, e.g. [12, 15, 16]. Other types of SGs include concurrent games [10, 31], limited information games [2, 13], and bidding games [3].

Stochastic systems with multiple objectives have been extensively studied for more than a decade. Markov decision processes, SGs with a single player, were investigated with multiple reachability or LTL objectives [24] as well as multiple discounted sum [14], total reward [29] or mean payoff objectives [18]. Further, the question of percentile queries was addressed in [28, 35], and combinations of probabilistic and non-probabilistic objectives in [7]. Non-standard multi-objective queries were developed together with domain experts in [5].

For SGs with multiple objectives, many decidability questions are still open. For conjunctive reachability, it is only known that the Pareto set can be approximated [1] with guaranteed precision, even in non-stopping games. For total reward, the problem is proven decidable only for stopping games with two-dimensional queries [9] but it can be approximated in higher dimensional stopping SGs [19]. If only deterministic strategies are allowed, the exact problem is undecidable [19], and so are generalized mean-payoff objectives in SGs [37]. However, keeping mean-payoff above a certain threshold with some probability is $coNP$ -complete [11]. Further, lexicographic preferences over multiple reachability or safety objectives can be reduced to single objectives [17]. The tool PRISM-games [32] implements compositional approaches to verification and strategy synthesis of several multi-objective problems [6, 33].

To the best of our knowledge, disjunctive queries were so far only considered as a special case of more general Boolean combinations for expected rewards [19], mean-payoff SGs [37], and in deterministic generalized reachability games on graphs [27].

Full version A full version of this paper including detailed proofs is available [38].

2 Preliminaries

General definitions. For sets A and B , the set of functions $A \rightarrow B$ is written B^A . The set of finite words over a non-empty set A is written A^* . For countable sets A we let $\text{Dist}(A) := \{P \in [0, 1]^A \mid \sum_{a \in A} P(a) = 1\}$ be the set of all *probability distributions* on A . The *support* of $d \in \text{Dist}(A)$ is defined

as $\text{supp}(d) = \{a \in A \mid d(a) > 0\}$. The i -th component of a vector $\vec{x} \in [0, 1]^n$ is denoted x_i . We compare vectors $\vec{x}, \vec{y} \in [0, 1]^n$ component-wise, i.e., $\vec{x} \leq \vec{y}$ iff $x_i \leq y_i$ for all $i = 1, \dots, n$. A set $X \subseteq \mathbb{R}^n$ is *convex* if for all $\vec{x}, \vec{y} \in X$ and $p \in [0, 1]$ it holds that $p\vec{x} + (1-p)\vec{y} \in X$. The *convex hull* $\text{conv}(X)$ of X is the smallest convex superset of X . Given sets $X, Y \subseteq \mathbb{R}^n$ and a real number $p \in [0, 1]$, we define the *p -convex combination* $pX + (1-p)Y := \{p\vec{x} + (1-p)\vec{y} \mid \vec{x} \in X, \vec{y} \in Y\}$. The *downward-closure* of $X \subseteq [0, 1]^n$ is defined as $\text{dwc}(X) := \{\vec{y} \in [0, 1]^n \mid \exists \vec{x} \in X: \vec{y} \leq \vec{x}\}$. X is called *downward-closed* if $X = \text{dwc}(X)$. A *closed half-space* is a set $\{\vec{x} \in \mathbb{R}^n \mid \vec{n} \cdot \vec{x} \geq d\}$ where $\vec{n} \in \mathbb{R}^n \setminus \{\vec{0}\}$ and $d \in \mathbb{R}$. A *polyhedron* is the intersection of finitely many closed half-spaces. Polyhedra are convex.

Stochastic games and strategies. Intuitively, the games considered in this paper are played by moving a pebble along the edges (called *transitions* from now on) of a finite directed graph. The vertices (subsequently called *states*) of this graph are partitioned into three classes which determine the states controlled by Eve, Adam, and the probabilistic environment, respectively:

Definition 1 (SG). A stochastic game (SG) is a tuple $\mathcal{G} = (S_E, S_A, S_P, s_0, P, \text{Act})$, where $S_E \uplus S_A \uplus S_P =: S$ are finite disjoint sets of states controlled either by Eve (S_E), Adam (S_A), or the probabilistic environment (S_P). The game starts in the initial state $s_0 \in S$. Each $s \in S_E \cup S_A$ has a non-empty set $\text{Act}(s) \subseteq S$ of actions available to Eve (Adam, resp.). For all $s \in S_P$, $P: S_P \rightarrow \text{Dist}(S)$ is a probability distribution over the successors of s .

For $s \in S_P$ and $t \in S$ we write $P(s, t)$ rather than $P(s)(t)$. A state $s \in S$ is called *sink* if either $s \in S_P$ and $P(s, s) = 1$ or $s \in S \setminus S_P$ and $\text{Act}(s) = \{s\}$. P and Act together induce a directed graph on S . We often sketch this *game graph* in our figures (e.g. Figure 1), drawing Eve's, Adam's and the probabilistic states with boxes, diamonds and circles, respectively; and we omit the self-loops on sinks to ease the presentation. For $k \geq 0$, we let $\mathcal{G}^{\leq k}$ be the restriction of \mathcal{G} to k -steps, that is, $\mathcal{G}^{\leq k}$ is obtained from \mathcal{G} by counting the transitions from s_0 taken so far and entering an error sink once their number exceeds k . A *Markov decision process* (MDP) is the 1-player version of an SG, i.e., either $S_E = \emptyset$ or $S_A = \emptyset$. A *Markov chain* (MC) is a 0-player SG, i.e., $S = S_P$. For technical reasons, we allow MDPs and MCs with countably infinite state spaces. SGs, on the other hand, are always *finite* in this paper.

Strategies define the semantics of SGs. A (general) strategy for Eve is a function $\sigma: S^*S_E \rightarrow \text{Dist}(S)$ such that $\text{supp}(\sigma(\pi s)) \subseteq \text{Act}(s)$ for all $\pi s \in S^*S_E$. A strategy σ is *deterministic* if $\sigma(\pi s)$ is a point-distribution for all $\pi s \in S^*S_E$, i.e., if it is not randomized.

To describe strategies by finite means (if possible), we use *strategy automata*. Formally, a strategy automaton for Eve is a structure $\mathcal{M} = (M, \mu, \nu, m_0)$ with M a countable set of memory elements, $\mu: S \times M \rightarrow M$ a *memory update* function, $\nu: S_E \times M \rightarrow \text{Dist}(S)$ a *next move* function, and $m_0 \in M$ an initial memory state. Given a strategy automaton \mathcal{M} , the *induced MDP* is the game

$$\mathcal{G}^{\mathcal{M}} := (\emptyset, S_A \times M, (S_P \cup S_E) \times M, (s_0, m_0), P^{\mathcal{M}}, \text{Act}^{\mathcal{M}}),$$

where the transition probability function $P^{\mathcal{M}}$ is defined as follows: Let $m, m' \in M$ be arbitrary and let $s \in S$. Then, if $s \in S_P$, we let $P^{\mathcal{M}}((s, m), (s', m')) := P(s, s')$ if $\mu(s', m) = m'$; if $s \in S_E$, then $P^{\mathcal{M}}((s, m), (s', m')) := \nu((s, m))(s')$ if $\mu(s', m) = m'$; and $P^{\mathcal{M}}((s, m), (s', m')) = 0$ in all other cases. Moreover, $\text{Act}^{\mathcal{M}}((s, m)) = \{(s', m') \mid s' \in \text{Act}(s) \wedge \mu(s', m) = m'\}$ for all $(s, m) \in S_A \times M$. \mathcal{M} is called a *stochastic-update* strategy if the memory update μ may additionally randomize over M (see [8] for details). From the definition it is clear that \mathcal{M} *realizes* a strategy in the general form $\sigma: S^*S_E \rightarrow \text{Dist}(S)$. We define the *memory size* of σ as the smallest $k \in \mathbb{N} \cup \{\infty\}$ such that there exists a strategy automaton $\mathcal{M} = (M, \mu, \nu, m_0)$ with $|M| = k$ that realizes σ . If $k = \infty$, then σ is *infinite-memory*, and otherwise *finite-memory*. If $k = 1$, then σ is called *memoryless*. An *MD* strategy is both memoryless and deterministic. The above definitions are analogous for the other player Adam, interchanging S_E and S_A .

Throughout the paper, we consistently denote Eve’s strategies with σ and Adam’s strategies with τ . We usually identify strategies with their realizing automata. Given a strategy σ , a *counter-strategy* τ is a strategy in the induced MDP \mathcal{G}^σ and may depend on σ .

Reachability-safety queries and determinacy. Given an MC (S, s_0, P) and a set $T \subseteq S$, the *reachability probability* of T is $\mathbb{P}(\diamond T) := \sum_{\pi \in \text{Paths}(T)} Pr(\pi)$ where $\text{Paths}(T)$ is the set of finite paths π of the form $\pi = s_0 s_1 \dots s_k$ with $k \geq 0$, $s_k \in T$ and $s_i \notin T$ for all $i = 0, \dots, k-1$; and $Pr(\pi) := \prod_{i=0}^{k-1} P(s_i, s_{i+1})$. Dually, we define $\mathbb{P}(\square T) := 1 - \mathbb{P}(\diamond \bar{T})$, where $\bar{T} := S \setminus T$. Intuitively, $\mathbb{P}(\diamond T)$ is the probability to eventually reach a *target* state in T and $\mathbb{P}(\square T)$ is the probability to stay forever within T , i.e., to avoid the *unsafe set* \bar{T} . We write $\mathbb{P}^{\sigma, \tau}$ to emphasize that we consider the probability measure in the Markov chain $\mathcal{G}^{\sigma, \tau}$ induced by some strategies σ, τ .

Definition 2 (Disjunctive Queries [19]). *Given an SG \mathcal{G} with state space S , an n -dimensional disjunctive query (DQ) for \mathcal{G} is an expression of the form $\varphi = \bigvee_{i=1}^n \mathbb{P}(\star_i T_i) \geq x_i$ with $\star_i \in \{\diamond, \square\}$, $T_i \subseteq S$ for all $i = 1, \dots, n$, and $\vec{x} \in (0, 1]^n$ a threshold vector.*

Note that we only allow (non-strict) *lower* bounds in DQs. This is w.l.o.g. as upper bounds on reachability or safety can be recast as lower bounds on the dual objective. The standard semantics of a DQ is defined as follows [19]: Eve can *achieve*¹ φ , or equivalently, φ is *achievable* in \mathcal{G} if $\exists \sigma \forall \tau: \bigvee_{i=1}^n \mathbb{P}^{\sigma, \tau}(\star_i T_i) \geq x_i$ where σ ranges over all strategies of Eve, τ over those of Adam and $\mathbb{P}^{\sigma, \tau}$ is the probability measure of the induced Markov chain $\mathcal{G}^{\sigma, \tau}$. A strategy of Eve witnessing achievability of φ is called an *achieving strategy*. Note that the quantification order is such that Eve has to reveal her strategy to Adam *before* the game actually starts. Since the games are not determined, this may be a disadvantage (see [19] or our example in the introduction). Therefore, we also consider the alternative semantics obtained by swapping the quantification order. We call this semantics *asserted-exposure* ($\forall \exists$ for short). In the $\forall \exists$ -semantics, Adam’s strategy is *exposed* to Eve *before* the game begins, i.e., φ is $\forall \exists$ -achievable if $\forall \tau \exists \sigma: \bigvee_{i=1}^n \mathbb{P}^{\sigma, \tau}(\star_i T_i) \geq x_i$ holds. By definition, \mathcal{G} is *determined* for φ iff the standard and the alternative $\forall \exists$ -semantics coincide.

We will consider the following subclasses of DQs: If $\star_i = \diamond$ ($\star_i = \square$) for all $i = 1, \dots, n$, then we call φ a *reachability (safety) DQ*. We say that φ is *mixed* to emphasize that it may contain both \diamond and \square . If $\vec{x} = (1, \dots, 1)$ then φ is called *qualitative DQ*, and otherwise *quantitative DQ*. If each state contained in a target/unsafe set is a sink then φ is called a *sink DQ*. All of the above notions are defined analogously for *conjunctive queries* (CQs).

Pareto sets. If the threshold vector \vec{x} in a query φ has not been fixed, we can think of φ as a *query template*. We define the set *Pareto set* for query template φ in a given SG \mathcal{G} as

$$\mathcal{A}(\mathcal{G}, \varphi) := \{ \vec{x} \in [0, 1]^n \mid \exists \sigma \forall \tau: \bigvee_{i=1}^n \mathbb{P}^{\sigma, \tau}(\star_i T_i) \geq x_i \}$$

and similarly for CQs. Further, for $k \geq 0$, we call $\mathcal{A}(\mathcal{G}^{\leq k}, \varphi)$ the *horizon- k Pareto set*, i.e., the set of points achievable if the game runs for at most k steps. We define $\mathcal{A}^{\forall \exists}(\mathcal{G}, \varphi)$ as the set of vectors achievable in the alternative $\forall \exists$ -semantics. Note that in general, $\mathcal{A}(\mathcal{G}, \varphi) \subseteq \mathcal{A}^{\forall \exists}(\mathcal{G}, \varphi)$, and equality holds iff \mathcal{G} is determined for all $\varphi(\vec{x})$, $\vec{x} \in [0, 1]^n$. The Pareto sets $\mathcal{A}(\mathcal{G}, \varphi)$ and $\mathcal{A}^{\forall \exists}(\mathcal{G}, \varphi)$ generalize the notion of *lower* and *upper value* in single-dimensional games. Indeed, they coincide for $n = 1$ as single-dimensional SGs are determined [22]. Furthermore, $\mathcal{A}(\mathcal{G}, \varphi)$ is convex for CQs [19].

¹We use the term “achieve” rather than “win” for consistency with previous works, e.g. [24, 19, 9].

Table 1: Memory requirements for Eve in terms of the number n of objectives in the DQ. The lower bounds in column “*SG with deterministic strats.*” apply already to sink queries.

	<i>SG with general strats.</i>	<i>SG with deterministic strats.</i>	<i>Non-SG with deterministic strats.</i>	
	\square / \diamond	\square / \diamond	\square	\diamond
<i>Qual.</i>	none [Lem. 2]	$\geq \binom{n}{\lfloor n/2 \rfloor}$ [Lem. 6]	$\binom{n}{\lfloor n/2 \rfloor}$ [27]	none [trivial]
<i>Quant.</i>	$2^n - 1$ [Lem. 1, 4, 5]	∞ [Cor. 1]	—not applicable—	

Goal-unfolding. The following construction is folklore (e.g. [27, 19]). Given an SG \mathcal{G} with states S and an n -dimensional query φ , we define the *goal-unfolding* $\text{Unf}(\mathcal{G}, \varphi)$ of \mathcal{G} with respect to φ as a game with state space $S \times \{0, 1\}^n$. Let (s, \vec{v}) be a state of the unfolding. Intuitively, $\text{Unf}(\mathcal{G}, \varphi)$ remembers which targets/unsafe sets have already been visited during a specific play. This is encoded in the n -bit vector \vec{v} . That is, if \mathcal{G} transitions from s to t , then in the unfolding (s, \vec{v}) moves to (t, \vec{u}) where \vec{u} is obtained from \vec{v} by setting all bits corresponding to the targets/unsafe sets containing t to one. Accordingly, the initial state is $(s_0, (0, \dots, 0))$. If φ is a sink query then the unfolding is trivial, i.e., equal to \mathcal{G} . Strategies in $\text{Unf}(\mathcal{G}, \varphi)$ can be interpreted as strategies in \mathcal{G} by incorporating the vectors \vec{v} from the states of the unfolding into the memory of the strategy.

3 Strategy Complexity

In this section we analyze the memory complexity of Eve’s achieving strategies for DQs in terms of the query dimension, denoted n in the following. More formally, given a class Q_n of DQs with n objectives, we determine (or bound) a number $b(n)$ such that

1. $b(n)$ memory is *necessary* for the queries in Q_n , i.e., there exists a game \mathcal{G} and query $\varphi \in Q_n$ such that φ is achievable in \mathcal{G} iff Eve may use at least $b(n)$ memory;
2. $b(n)$ memory is *sufficient* for the queries in Q_n , i.e., for all games \mathcal{G} and $\varphi \in Q_n$, if φ is achievable in \mathcal{G} , then Eve has an achieving strategy using at most $b(n)$ memory.

To give a nuanced picture of the complexity, we distinguish the classes of qualitative vs. quantitative DQs, safety vs. reachability DQs and study the restriction to deterministic vs. general (randomized) strategies. We stress that the latter distinction applies to both players, i.e., in the *deterministic-strategies* case, neither Adam nor Eve may play randomized strategies whereas in the *general-strategies* case, both players may follow arbitrary—possibly randomized—strategies.

We briefly recall the case of non-stochastic games with deterministic strategies for both players. It was shown in [27, Lem. 1] that $\binom{n}{\lfloor n/2 \rfloor}$ memory is necessary and sufficient for safety DQs. Notably, this means that not the *whole* goal-unfolding is needed (though an exponentially large fragment). Reachability DQs, on the other hand, do not need memory at all in the purely deterministic setting because a DQ $\bigvee_{i=1}^n \mathbb{P}(\diamond T_i) \geq 1$ boils down to reaching the set $\bigcup_{i=1}^n T_i$, and MD strategies are sufficient for winning reachability games on finite graphs.

In the rest of the section we first treat the general-strategies case (Section 3.1) and then study the restriction to deterministic strategies (Section 3.2). Table 1 summarizes the results.

Theorem 1. *The memory bounds in Table 1 are correct.*

3.1 Strategy Complexity under General Strategies

Recall from above that whenever we say that a certain class of strategies are “sufficient” or “necessary”, we are implicitly assuming that Eve actually has a winning strategy.

Lemma 1. *In the general-strategies case, deterministic strategies with at most $2^n - 1$ memory states suffice for mixed quantitative DQs. Moreover, MD strategies are sufficient for sink queries.*

Proof (sketch). By [19, Theorem 7], MD strategies are sufficient for quantitative DQs with *expected reward* objectives (see the formal definition in [19]). We reduce reachability and safety to expected reward in the goal-unfolding [38]. The resulting MD strategy in the unfolding corresponds to a strategy of Eve with at most $2^n - 1$ memory. \square

Recall that even though Lemma 1 implies that deterministic strategies are sufficient for Eve, Adam may still use randomization. In fact, we show in Corollary 1 in Section 3.2 that Lemma 1 does *not* hold in the deterministic-strategies case where both players are forced to follow a deterministic strategy. Next we either improve the bound from Lemma 1 or prove matching lower bounds. We consider qualitative DQs first.

Lemma 2. *In the general-strategies case, MD strategies suffice for mixed qualitative DQs.*

Proof (sketch). It can be shown that a given strategy of Eve achieves $\bigvee_{i=1}^n \mathbb{P}(\star_i T_i) \geq 1$ iff it achieves the single objective $\mathbb{P}(\star_i T_i) \geq 1$ for at least one $1 \leq i \leq n$ [38]. For the latter, MD strategies are sufficient [21]. \square

The qualitative bounds (“ ≥ 1 ”) are crucial in the previous proof. Indeed, the equivalence in the above proof sketch does *not* hold for quantitative bounds (a minimal counter-example is the game in Fig. 1 started from s_2 with bounds $\geq 1/2$ for both objectives). For the quantitative setting where arbitrary bounds are allowed, the following observation relates conjunctive and disjunctive queries and enables us to reuse some known results about CQs. Intuitively, it states that qualitative reachability CQs can be reduced to quantitative (qualitative) DQs in the case of general (respectively deterministic) strategies.

Lemma 3. *For every SG \mathcal{G} with a qualitative reachability CQ $\varphi = \bigwedge_{i=1}^n \mathbb{P}(\diamond T_i) \geq 1$, there exists a game \mathcal{G}' (where Eve’s strategies σ are in one-to-one correspondence) and a*

1. quantitative reachability DQ φ'_{quant} such that σ achieves φ in \mathcal{G} iff σ achieves φ'_{quant} in \mathcal{G}' under general strategies;
2. qualitative reachability DQ φ'_{qual} such that σ achieves φ iff σ achieves φ'_{qual} under deterministic strategies.

Proof (sketch). \mathcal{G}' is constructed as follows: The original \mathcal{G} is only played with probability $1/2$. With the remaining $1/2$, Adam freely chooses one of the n targets T_1, \dots, T_n , after which the game ends immediately (Figure 2, left). The following can be readily verified (see [38]): In the deterministic-strategies case, each strategy σ of Eve achieves the CQ φ in \mathcal{G} iff σ achieves the *qualitative* DQ $\bigvee_{i=1}^n \mathbb{P}(\diamond T_i) \geq 1$ in \mathcal{G}' . Otherwise, if randomization is allowed, then σ achieves φ in \mathcal{G} iff it achieves the *quantitative* DQ $\bigvee_{i=1}^n \mathbb{P}(\diamond T_i) \geq \frac{1}{2} + \frac{1}{2n}$ in \mathcal{G}' . \square

With Lemma 3 and a result of [27] we obtain a lower bound for quantitative reachability:

Lemma 4. *In the general-strategies case, quantitative reachability DQs need $2^n - 1$ memory.*

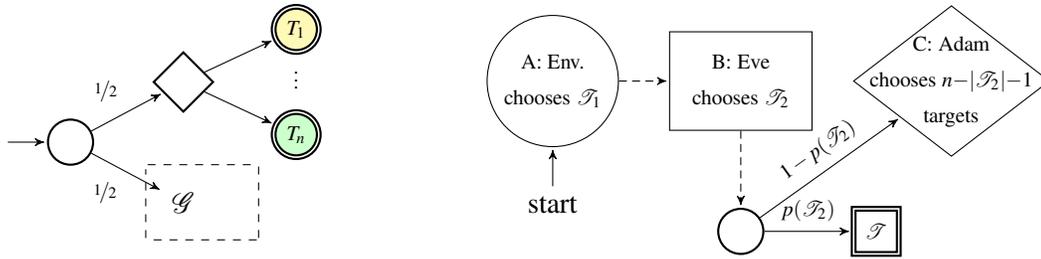


Figure 2: Left: Reduction of qualitative CQs to DQs (Lemma 3). Right: Game constructed in the proof of Lemma 5. The probability $p(\mathcal{T}_2)$ increases with $|\mathcal{T}_2|$. Eve must select exactly $\mathcal{T}_2 = \mathcal{T}_1$ in Stage B to avoid visiting at least one of the targets in \mathcal{T} with maximal probability.

Proof (sketch). [27, Lem. 2] describes a family of (deterministic) games $(\mathcal{G}_n)_{n \geq 1}$ where Eve needs $2^n - 1$ memory to visit all n targets under *deterministic* strategies, i.e., to achieve a *qualitative reachability CQ*. A more detailed analysis of the games \mathcal{G}_n shows that even if both players may use general strategies, Eve still needs $2^n - 1$ memory [38]. We stress that this is non-trivial: in some cases, memory *can* be traded for randomization to achieve qualitative DQs. A minimal example is an MDP with $S_E = \{s_0\}$, $S_P = \{t_1, t_2\}$, $P(t_1, s_0) = P(t_2, s_0) = 1$, $\text{Act}(s_0) = \{t_1, t_2\}$, $T_1 = \{t_1\}$, $T_2 = \{t_2\}$; the CQ $\mathbb{P}(\diamond T_1) \geq 1 \wedge \mathbb{P}(\diamond T_2) \geq 1$ is achievable by a memoryless randomized strategy but not by an MD strategy. By Lemma 3.1, we can reduce each \mathcal{G}_n to an SG with a *quantitative reachability DQ* where Eve’s winning strategies are the same as those in the original \mathcal{G} , i.e., they require at least $2^n - 1$ memory. \square

We now prove a lower bound for safety DQs. To the best of our knowledge, unlike Lemma 4, the bound for safety DQs does not easily follow from related works. Therefore we propose a new construction for this case (shown on the right of Figure 2; see [38] for the proof).

Lemma 5. *In the general-strategies case, quantitative safety DQs need $2^n - 1$ memory.*

In summary, quantitative DQs require the *full* goal-unfolding—which is of exponential size in n —while qualitative ones do not need memory at all.

3.2 Strategy Complexity under Deterministic Strategies

Example 1. *Consider the SG in Figure 3 (left). The qualitative DQ $\mathbb{P}(\diamond T_1) \geq 1 \vee \mathbb{P}(\diamond T_2) \geq 1$ is achievable if Adam has to choose deterministically in s_0 : If he chooses the upper path, then Eve chooses T_1 in s_1 ; conversely, if he chooses the lower path, then Eve moves to T_2 . Clearly, this strategy requires Eve to use one bit of memory (i.e., memory size 2) and there is no memoryless strategy that achieves the query. This example shows that Lemma 1 is not valid in the deterministic-strategies case: memory is needed even for sink queries.*

The above example indicates that the deterministic-strategies case is already “interesting” for sink queries. Therefore—due to space limitations of the paper—we have decided to restrict our study of deterministic strategies to sink queries only. The somewhat counter-intuitive situation that *more* memory is needed if *less general* strategies are allowed is due to the fact that only Adam benefits from randomization, but not Eve. In fact, Eve can in general achieve *more* queries in the deterministic-strategies setting. Nonetheless, the example in Figure 3 shows that Eve’s added power does not come for free: She has to invest more resources (memory) on her side as well. The following shows that this is necessary in general:

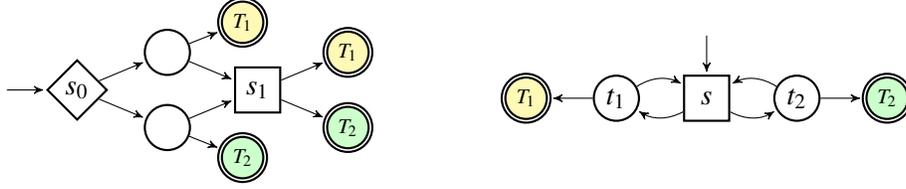


Figure 3: All probabilities are 0.5. Left: If only deterministic strategies are allowed, then Eve’s achieving strategies for $\mathbb{P}(\diamond T_1) \geq 1 \vee \mathbb{P}(\diamond T_2) \geq 1$ need memory, despite the fact that targets/unsafe sets contain only sinks. Right: There exists x such that the CQ $\mathbb{P}(\diamond T_1) \geq x \wedge \mathbb{P}(\diamond T_2) \geq 1-x$ is achievable under deterministic strategies iff infinite memory is allowed.

Lemma 6. *In the deterministic-strategies case, $\binom{n}{n/2}$ memory is necessary for qualitative reachability or safety DQs. This holds even for sink queries.*

Proof (sketch). We sketch the construction for qualitative reachability DQs. Let $n = 2q$ be even. The game comprises three stages, A, B, and C. In the initial Stage A, Adam specifies a combination \mathcal{T}_A of q different targets. Those are then visited with probability $1/2$ and the game ends. With the remaining probability of $1/2$, the game moves on to Stage B which is similar to stage A, but controlled by Eve. Let \mathcal{T}_B be the set of q targets that Eve specifies in this stage. Consequently, with total probability $1/4$, the game enters the final stage C where Adam chooses and visits $q+1$ different targets \mathcal{T}_C . It follows that a target is visited with probability 1 iff it is chosen in all three stages. The only achieving strategy of Eve consists in selecting exactly $\mathcal{T}_B = \mathcal{T}_A$, requiring $\binom{n}{q}$ memory. See [38] for the remaining details and the adaptation of the construction to safety. \square

We consider quantitative DQs next. The following lemma is the quantitative-bounds version of Lemma 3. The difference is, however, that the reduction only works under *deterministic* strategies (see [38] for the proof).

Lemma 7. *For every SG \mathcal{G} with quantitative reachability CQ $\varphi = \bigwedge_{i=1}^n \mathbb{P}(\diamond T_i) \geq x_i$, there exists a game \mathcal{G}' (where Eve’s strategies σ are in one-to-one correspondence) and a quantitative reachability DQ φ' such that, under the assumption of deterministic strategies, σ achieves φ in \mathcal{G} iff σ achieves φ' in \mathcal{G}' .*

We can use the previous reduction to show that in general, infinite memory is necessary for achieving quantitative DQs under deterministic strategies.

Lemma 8. *For the MDP from Figure 3 (right), there exists $x \in [0, 1]$ such that the CQ $\mathbb{P}(\diamond T_1) \geq x \wedge \mathbb{P}(\diamond T_2) \geq 1-x$ is only achievable by an infinite-memory strategy.*

Proof. Every deterministic strategy σ can be identified with an infinite string $\sigma = \sigma_1 \sigma_2 \dots \in \{0, 1\}^\omega$ where for all $i \geq 1$, $\sigma_i = 1$ ($\sigma_i = 0$) indicates that σ moves to t_1 (t_2 , resp.) when s is entered for the i -th time. Clearly, $\mathbb{P}^\sigma(\diamond T_1) = (0.\sigma)_2$, i.e., the probability to reach T_1 is equal to the real number whose decimal binary representation is the infinite string $0.\sigma$. We claim that the above CQ $\mathbb{P}(\diamond T_1) \geq x \wedge \mathbb{P}(\diamond T_2) \geq 1-x$ with $x = (0.1^1 01^2 01^3 01^4 0\dots)_2$ can only be achieved by a strategy that uses infinite memory. If not, then let \mathcal{M} be a finite-state strategy automaton that achieves the CQ. Since \mathcal{M} is finite, there exist two distinct prefixes $\pi_1 \neq \pi_2$ of $1^1 01^2 01^3 01^4 0\dots$ such that after reading π_1 or π_2 , the automaton \mathcal{M} is in the same memory state m . Suppose that from m on, \mathcal{M} plays action sequence $\pi \in \{0, 1\}^\omega$. Since \mathcal{M} is achieving, we must have that $\pi_1 \pi = \pi_2 \pi = 1^1 01^2 01^3 01^4 0\dots$ which, however, implies $\pi_1 = \pi_2$, contradiction. \square

Table 2: Complexity of the feasibility problem for DQs in the standard semantics. Column “SG with deterministic strats.” applies exclusively to sink queries.

	<i>SG with general strats.</i>	<i>SG with deterministic strats.</i>	<i>Non-SG with deterministic strats.</i>	
	\square / \diamond	\square / \diamond	\square	\diamond
<i>Qual.</i>	P [Lem. 9]	\geq PSPACE [Lem. 12] \leq EXPTIME [Lem. 13] / ?	PSPACE [27]	P [trivial]
<i>Quant.</i>	\geq PSPACE [Lem. 11,10] \leq NEXPTIME	undecidable [Lem. 14]	—not applicable—	

Corollary 1. *In general, infinite memory is necessary for achieving quantitative reachability or safety DQs under deterministic strategies. This holds even for sink queries.*

Proof. Apply the reduction from Lemma 7 to the MDP from Lemma 8. For safety notice that in the MDP, $\mathbb{P}(\square \bar{T}_1) \geq x$ iff $\mathbb{P}(\diamond T_2) \geq x$. \square

4 Computational Complexity

In this section, we study the complexity of the achievability problem for DQs in the standard semantics, i.e., the decision problem “ $\exists \sigma \forall \tau: \bigvee_{i=1}^n \mathbb{P}^{\sigma, \tau}(\star_i) \geq x_i$ ” in some given game. We consider the same variations of the problem as in Section 3, that is, qualitative vs. quantitative DQs, reachability vs. safety queries and deterministic vs. general strategies. For the complexity theoretic results, we assume that all transition probabilities in the games and thresholds in the queries are rational numbers given as binary-encoded integer pairs.

We again briefly discuss the case of purely deterministic games on graphs. [27, Theorem 1] shows that, in deterministic games, qualitative safety DQs are PSPACE-complete via a reduction from quantified Boolean formulas. Reachability DQs, as mentioned in Section 3.1, can be reduced to solving a standard single-target reachability game which can be solved in P. We now present our results in detail, first for general (Section 4.1) and then for deterministic strategies (Section 4.2). Table 2 summarizes the results.

Theorem 2. *The complexity bounds in Table 2 are correct.*

4.1 Computational Complexity under General Strategies

Lemma 9. *In the general-strategies case, qualitative mixed DQs are decidable in P.*

Proof. As in Lemma 2, in the qualitative case it suffices to check for each objective $\star_i T_i$, $i = 1, \dots, n$, $\star_i \in \{\diamond, \square\}$, individually whether Eve can satisfy it with probability 1. Hence n queries to a polynomial time algorithm for qualitative simple stochastic games [25] suffice. \square

Lemma 10. *In the general-strategies case, quantitative reachability DQs are PSPACE-hard.*

Proof. The feasibility problem for qualitative reachability CQs is PSPACE-hard, which holds already for MDPs where $S_A = \emptyset$ [35, Lem. 2]. Lemma 3.1 in Section 3.1, reduces the MDP CQ problem to a quantitative DQ problem in an SG in polynomial time. \square

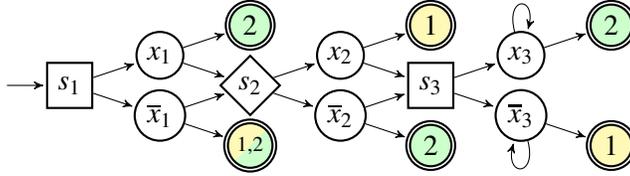


Figure 4: Example reduction from Lemma 12 for the QBF $\exists x_1 \forall x_2 \exists x_3 (\bar{x}_1 \wedge x_2 \wedge \bar{x}_3) \vee (\bar{x}_2 \wedge x_3)$. Numbers in target states indicate whether they belong to T_1 and/or T_2 . The QBF is true as witnessed by Eve’s strategy that first goes to \bar{x}_1 and then to x_3 if Adam had selected \bar{x}_2 and otherwise to \bar{x}_3 .

Lemma 11. *In the general-strategies case, quantitative safety DQs are PSPACE-hard.*

Proof (sketch). It can be shown that quantitative reachability CQs with *strict bounds* in MDPs can be reduced to quantitative safety DQs (with non-strict bounds) in SGs. We prove that strict-bounded reachability CQs in MDPs are PSPACE-hard which is done by analyzing a construction from [35, Lem. 2] in greater detail (see [38]). \square

PSPACE-hardness in the previous two lemmas is caused by the exponential size of the goal-unfolding that can, as we have shown in Section 3.1, not be avoided in general. Indeed, for sink queries the complexity of quantitative (mixed) DQs drops to NP-complete in the general-strategies case [19, Corollary 1], where the upper bound stems from the fact that MD strategies suffice and can be verified in polynomial time using linear programming [24].

Regarding upper bounds on quantitative DQ feasibility, we remark that the problem can be decided in NEXPTIME: Guess an (exponentially large) MD strategy σ in the goal unfolding, consider the induced MDP \mathcal{G}^σ and verify in polynomial time in the size of \mathcal{G}^σ that Adam does not have a strategy violating all thresholds at once by using the multi-objective MDP algorithm from [24]. We currently do not know of a tighter upper bound.

4.2 Computational Complexity under Deterministic Strategies

As in Section 3.2, we consider only sink queries in this section.

Lemma 12. *Under deterministic strategies, qualitative reachability and safety DQs are PSPACE-hard, even in the case of sink queries.*

Proof (sketch). Inspired by similar constructions in [27, 35], we reduce from the problem of deciding truth of a quantified Boolean formula (QBF). See Figure 4 for an example and [38] for the full proof. \square

Notably, the corresponding *conjunctive* problem in the setting of Lemma 12 (qualitative sink CQ, deterministic strategies) can be solved in P as it reduces to simply checking if the intersection $\bigcap_{i=1}^n T_i$ can be reached with probability 1. Contrary to most other results, Lemma 12 thus identifies a setting where DQs are *much* harder than CQs. Moreover, due to the restriction to sink queries, Lemma 12 also yields PSPACE-hardness for disjunctions of expected reward objectives under deterministic strategies. In the general-strategies case, such expected reward DQs are decidable in NP [19]. Next we show an upper bound for qualitative *safety* DQs:

Lemma 13. *In the deterministic-strategies case, qualitative safety sink DQs are decidable in EXPTIME.*

	$F(X)_s$	$\Phi(\mathcal{X})_s$
$s \in S_A$	$\bigcap_{t \in \text{Act}(s)} X_t$	$\bigcup_{t \in \text{Act}(s)} \mathcal{X}_t$
$s \in S_E$	$\text{conv}(\bigcup_{t \in \text{Act}(s)} X_t)$	$\{ \text{conv}(\bigcup_{t \in \text{Act}(s)} X_t) \mid X \in \mathcal{X} \}$
$s \in S_P$	$\sum_{t \in S} P(s, t) X_t$	$\{ \sum_{t \in S} P(s, t) X_t \mid X \in \mathcal{X} \}$

Figure 5: The value iteration operators F and Φ for computing $\mathcal{A}(\mathcal{G}^{\leq k}, \varphi)$ and $\mathcal{A}^{\forall\exists}(\mathcal{G}^{\leq k}, \varphi)$, respectively.

Proof (sketch). The proof relies crucially on the fact that non-achievability of a qualitative safety DQs can be witnessed after at most a bounded number steps of the game. Indeed, if σ is a non-achieving strategy of Eve, then Adam has a counter-strategy that can reach all the unsafe sets \bar{T} with *positive* probability after at most $|S|$ steps of the game. The result then follows by constructing a polynomially space-bounded alternating Turing machine that simulates the game for at most $|S|$ steps and accepts iff the query is not achievable. We handle probabilistic branching via backtracking using a stack whose content remains of polynomial size throughout the execution (see [38] for details). \square

The above proof cannot simply be extended to reachability because, intuitively, reaching a target with probability 1 may only occur in the limit. In fact, the question whether qualitative reachability DQs are decidable under deterministic strategies remains open.

Regarding the quantitative case, [19, Theorem 3] proves that reachability CQs are *undecidable* under deterministic strategies. Thus with Lemma 7 we also have:

Lemma 14. *Quantitative reachability DQs are undecidable under deterministic strategies.*

5 Value Iteration

In a nutshell, value iteration (VI) algorithms in general evaluate the k -step game $\mathcal{G}^{\leq k}$ using information about the game $\mathcal{G}^{\leq k-1}$ up to some reasonable k . In this section, we present a VI-style algorithm for computing the Pareto sets $\mathcal{A}(\mathcal{G}^{\leq k}, \varphi)$ or $\mathcal{A}^{\forall\exists}(\mathcal{G}^{\leq k}, \varphi')$ for a given game \mathcal{G} , DQ φ or CQ φ' , and step bound $k \geq 0$. Consequently, we do not fix a threshold vector \vec{x} in our queries but consider query *templates* instead. To keep the presentation simple, we focus on *general-strategies* and consider only sink queries.

We briefly recall the VI from [19] (subsequently called *CQ-VI*) that for a given CQ φ successively outputs $\mathcal{A}(\mathcal{G}^{\leq k}, \varphi)$ for all $k \geq 0$. Let \mathfrak{C} be the set of all downward-closed polyhedra in $[0, 1]^n$ and let $X \in \mathfrak{C}^S$. In the following, we write X_s for $X(s)$. The update function $F: \mathfrak{C}^S \rightarrow \mathfrak{C}^S$ is defined according to Figure 5. Note that F is well-defined, i.e., always yields downward-closed polyhedra. For CQ $\varphi = \bigwedge_{i=1}^n \mathbb{P}(\star_i T_i) \geq x_i$ and each $s \in S$ define the n -dimensional zero-one vector $\mathbb{1}_s^\varphi$, such that $(\mathbb{1}_s^\varphi)_i := 1$ iff $s \in T_i$ and let $X^0(\varphi)_s := \text{dwc}(\mathbb{1}_s^\varphi)$. Then [19] implies that $F^k(X^0(\varphi))_{s_0} = \mathcal{A}(\mathcal{G}^{\leq k}, \varphi)$ for all $k \geq 0$.

We now address the question whether an iteration analogous to CQ-VI can be devised for *DQs*. CQ-VI computes the horizon- $(k+1)$ Pareto set of any given state by taking only the horizon- k sets of its successors (and the relevant probability distribution) into account. For DQs, this is impossible in general:

Observation 1. *Suppose $s \in S$ has successors s_1, s_2 . In general, the horizon- k Pareto sets of s_1 and s_2 w.r.t. a DQ do not uniquely determine the horizon- $(k+1)$ Pareto set of s .*

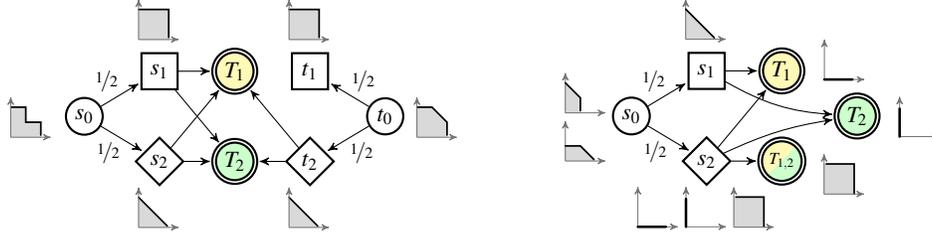


Figure 6: Left: The successors of both s_0 and t_0 have the same Pareto set w.r.t. the DQ $\mathbb{P}(\square\overline{T}_1) \geq x_1 \vee \mathbb{P}(\square\overline{T}_2) \geq x_2$, but the Pareto sets of s_0 and t_0 are different. Right: Example run of Algorithm 1 for $k = 2$ and query $\mathbb{P}(\diamond T_1) \geq x_1 \wedge \mathbb{P}(\diamond T_2) \geq x_2$. At s_2 , the rightmost polyhedron is removed by the μ -operation in line 4. The result is the intersection of the two polyhedra at s_0 .

Proof. Consider the game in Figure 6 (left) and the DQ $\varphi = \mathbb{P}(\square\overline{T}_1) \geq x_1 \vee \mathbb{P}(\square\overline{T}_2) \geq x_2$. The horizon-1 Pareto sets of s_1, s_2 and t_1, t_2 , as well as the horizon-2 sets of s_0 and t_0 are sketched next to the corresponding state. We claim that the threshold vector $(x_1, x_2) = (0.75, 0.75)$ is achievable from t_0 , but not from s_0 : As deterministic strategies suffice for Eve, we can assume by symmetry that she moves to T_1 in s_1 . But then Adam can respond by moving to T_2 in s_2 and both $\square\overline{T}_1$ and $\square\overline{T}_2$ are satisfied with probability exactly 0.5. Thus $(0.75, 0.75)$ is not achievable. However, at t_2 we can assume by symmetry that Adam moves to T_1 with probability ≥ 0.5 and so $\square\overline{T}_2$ is satisfied with probability ≥ 0.75 from t_0 . \square

Intuitively, the example in Figure 6 demonstrates that—unlike in CQ-VI—the Pareto sets alone do not convey enough information to allow for a sound VI. In the remainder of this section we present a workaround for this problem. The idea is to account for the missing information by extending the domain of VI to *sets* of Pareto sets. We will not work with DQs directly but with CQs in the $\forall\exists$ -semantics. This is justified because *non-achievability* of a DQ can be recast as follows:

$$\neg\exists\sigma\forall\tau\bigvee_{i=1}^n\mathbb{P}^{\sigma,\tau}(\star_i T_i) \geq x_i \iff \forall\sigma\exists\tau\bigwedge_{i=1}^n\mathbb{P}^{\sigma,\tau}(\overline{\star}_i \overline{T}_i) > 1 - x_i \quad (1)$$

where $\overline{\diamond} = \square$ and $\overline{\square} = \diamond$. That is, for deciding achievability of a DQ $\varphi = \bigvee_{i=1}^n \mathbb{P}(\star_i T_i) \geq x_i$, we can equivalently consider the dual CQ $\overline{\varphi} := \bigwedge_{i=1}^n \mathbb{P}(\overline{\star}_i \overline{T}_i) > 1 - x_i$ from (1) under the $\forall\exists$ -semantics in the game \mathcal{G} where the roles of Adam and Eve have been swapped. In fact, with (1), the whole Pareto set $\mathcal{A}(\mathcal{G}, \varphi)$ can be recovered from $\mathcal{A}^{\forall\exists}(\mathcal{G}, \overline{\varphi})$.

To define our VI, we introduce some auxiliary notation first. $\mathcal{P}(\mathcal{C})$ denotes the powerset of \mathcal{C} . For mappings $X \in \mathcal{C}^S$ and $\mathcal{X} \in \mathcal{P}(\mathcal{C})^S$ from states to (sets of) polyhedra, we write $X \dot{\in} \mathcal{X}$ iff $X(s) \in \mathcal{X}(s)$ for all $s \in S$. Further, we let $\{X\}$ be the lifting of X to $\mathcal{P}(\mathcal{C})^S$, i.e., $\{X\}_s := \{X_s\}$. Formally, for any fixed CQ φ our new VI can be seen as a function $\Phi: \mathcal{P}(\mathcal{C})^S \rightarrow \mathcal{P}(\mathcal{C})^S$ and is defined according to Figure 5. The iteration is started with $\mathcal{X}^0(\varphi) := \{X^0(\varphi)\}$, where $X^0(\varphi)$ is the same initial element as for CQ-VI.

Lemma 15. *For all $k \geq 0$ we have that $\bigcap (\Phi^k(\mathcal{X}^0)_{s_0}) = \mathcal{A}^{\forall\exists}(\mathcal{G}^{\leq k}, \varphi)$.*

Proof (sketch). It can be shown that for all $k \geq 0$, the set $\Phi^k(\mathcal{X}^0)_{s_0}$ contains the Pareto sets achievable by Eve in the k -step MDPs induced by each possible *deterministic* k -step strategy τ of Adam [38]. The final intersection over $\Phi^k(\mathcal{X}^0)_{s_0}$ is due to the outer \forall -quantifier in the $\forall\exists$ -semantics. \square

The iteration according to Lemma 15 is essentially equivalent to enumerating all possible deterministic k -step strategies of Adam and analyzing the induced MDPs. In the worst case, there are doubly

exponentially many (in k) such strategies and thus the number of polyhedra $|\Phi^k(\mathcal{X}^0)_s|$ maintained per state $s \in S$ in the k -th step is also at most doubly exponential. This can be improved. In general, not all k -step strategies have to be considered: If for k -step strategies τ, τ' it holds that the set of points achievable by Eve in the induced MDP \mathcal{G}^τ is contained in set of points achievable in $\mathcal{G}^{\tau'}$, then only τ is relevant and τ' can be discarded. Intuitively, Adam would always (independently of the thresholds \bar{x}) prefer τ over τ' in such a situation. We can incorporate this observation into our value iteration: Let $\mu: \mathcal{P}(\mathcal{C})^S \rightarrow \mathcal{P}(\mathcal{C})^S$ be the function that removes the non-inclusion-minimal polyhedra of each \mathcal{X}_s . We can then iterate $\mu \circ \Phi$ instead of Φ without changing the result of the “final intersection” in Lemma 15. We summarize the overall procedure as Algorithm 1.

Theorem 3. *Algorithm 1 is correct.*

Experiments. To assess the complexity of Algorithm 1 in practice we have built a prototypical implementation² using the PARMA POLYHEDRA LIBRARY [4]. We have tested our implementation on a variant of the smart heating example from [9] that was itself inspired from the case study in [34]. Further, we consider randomly generated 2-dimensional games with 10 states (see [38] for more details). To determine the relative overhead of our algorithm compared to CQ-VI we consider the number n_s^k of polyhedra maintained at state s in iteration k . For the floor heating example we found that $n_s^k = 1$ for all $k \geq 0$ and $s \in S$, which means that Adam has a unique optimal strategy from each s and step-bound k . Moreover, the game is determined. For the randomly generated games, we observed that approximately 90% of them also had $n_s^k = 1$ for all $k \geq 0$ and $s \in S$. We conjecture that this is indeed a typical situation (as in the floor heating example), however, 90% might be a too high estimate due to trivial random games. In the following, we only consider “hard” instances with $n_s^k > 1$ for at least one state s and some $k \geq 0$. In Table 4, we report the mean number of polyhedra $\bar{n}^k = \frac{1}{|S|} \sum_{s \in S} n_s^k$ of 100 “hard” games for various iteration counts k . The empirical average of \bar{n}^k over the 100 instances that were processed within the timeout is given in column $E[\bar{n}^k]$ and the number of timeouts (10 seconds) in column T/O. We have also compared the algorithm with and without the μ -operation in Line 4 of Algorithm 1 (columns $\mu \circ \Phi$ and Φ , respectively).

In summary, our experiments show that in many cases the necessary number of polyhedra is low enough to be feasible, often even only 1. In “hard” cases, our results show that after dozens of iterations the number of polyhedra blows up dramatically, frequently resulting in a timeout (which is why the numbers for Φ decrease after 10 iterations, as only the instances with lower \bar{n}^k finish). This highlights the difficulty of DQs compared to CQs. Still, using our optimization μ , in many “hard” cases the computation finishes and the number of polyhedra per state stays below 10, and thus is 2 orders of magnitude smaller than without μ .

6 Conclusion and Future Work

We have presented a detailed picture of computational and strategy complexity of SGs with DQ winning conditions. The results were obtained in part by providing reductions from CQs to DQs and applying results from the literature. Future work on the complexity side includes closing the gaps in Tables 1 and 2; however, we conjecture that this requires significant new insights. For example, a major obstacle towards proving PSPACE membership of the quantitative general-strategies DQs problem is that one has to reason about *exact* reachability probabilities in the exponentially large goal-unfolding. It is not

²Available at <https://doi.org/10.5281/zenodo.5047440>

Algorithm 1: Value Iteration for CQs in the asserted-exposure ($\forall\exists$) semantics.

Input: Game \mathcal{G} , (mixed) CQ φ , horizon $k \geq 0$
Output: The horizon- k Pareto set $\mathcal{X}^{\forall\exists}(\mathcal{G}^{\leq k}, \varphi)$

- 1: $\mathcal{X} \leftarrow \{X^0(\varphi)\}$ ▷ Initialization
- 2: **for** i from 1 to k **do**
- 3: $\mathcal{X} \leftarrow \Phi(\mathcal{X})$ ▷ Apply Φ according to Figure 5
- 4: $\mathcal{X} \leftarrow \mu(\mathcal{X})$ ▷ Keep only \subseteq -minima of each \mathcal{X}_s
- 5: **return** $\bigcap \mathcal{X}_{s_0}$ ▷ Intersection of curves at initial state

Table 3: Experimental results for a fixed timeout of 10s.

k	$E[\bar{\pi}^k]$		T/O	
	Φ	$\mu \circ \Phi$	Φ	$\mu \circ \Phi$
1	1.2	1.1	0	0
5	16.1	1.8	0	0
10	526.3	6.8	63	12
20	202.4	5.4	80	30
100	78.9	2.6	90	50

at all obvious that the number of bits needed for the rational representations of these quantities remains polynomially bounded.

We have also argued that DQs are equivalent to CQs in the optimistic “asserted-exposure” ($\forall\exists$) semantics obtained by changing the quantification order over strategies—unlike in simple SGs, this makes a difference since our games are not always determined. Moreover, we have formulated the first VI-style algorithm for DQs in the standard and CQs in the $\forall\exists$ -semantics. It should be straightforward to extend our algorithm to expected rewards as well. Another interesting application of the algorithm is to certify determinacy (for a finite step bound). Regarding future work, it would be appealing to implement the algorithm in a tool such as PRISM-games and to experiment with more realistic case studies. Yet another direction is to investigate (counter-)strategy synthesis for Eve in the $\forall\exists$ -semantics, e.g., by constructing strategy templates where some choices depend on Adam’s observable strategy.

References

- [1] Pranav Ashok, Krishnendu Chatterjee, Jan Kretínský, Maximilian Weininger & Tobias Winkler (2020): *Approximating Values of Generalized-Reachability Stochastic Games*. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi & Dale Miller, editors: *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, ACM, pp. 102–115, doi:10.1145/3373718.3394761.
- [2] Pranav Ashok, Jan Kretínský & Maximilian Weininger (2019): *PAC Statistical Model Checking for Markov Decision Processes and Stochastic Games*. In Isil Dillig & Serdar Tasiran, editors: *Computer Aided Verification - 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I, Lecture Notes in Computer Science 11561*, Springer, pp. 497–519, doi:10.1007/978-3-030-25540-4_29.
- [3] Guy Avni & Thomas A. Henzinger (2020): *A Survey of Bidding Games on Graphs (Invited Paper)*. In Igor Konnov & Laura Kovács, editors: *31st International Conference on Concurrency Theory, CONCUR 2020, September 1-4, 2020, Vienna, Austria (Virtual Conference), LIPIcs 171*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 2:1–2:21, doi:10.4230/LIPIcs.CONCUR.2020.2.
- [4] Roberto Bagnara, Patricia M. Hill & Enea Zaffanella (2008): *The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems*. *Sci. Comput. Program.* 72(1-2), pp. 3–21, doi:10.1016/j.scico.2007.08.001.
- [5] Christel Baier, Clemens Dubslaff, Sascha Klüppelholz, Marcus Daum, Joachim Klein, Steffen Märcker & Sascha Wunderlich (2014): *Probabilistic Model Checking and Non-standard Multi-objective Reasoning*. In Stefania Gnesi & Arend Rensink, editors: *Fundamental Approaches to Software Engineering - 17th Inter-*

- national Conference, FASE 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings, Lecture Notes in Computer Science 8411, Springer, pp. 1–16, doi:10.1007/978-3-642-54804-8_1.
- [6] Nicolas Basset, Marta Z. Kwiatkowska & Clemens Wiltsche (2014): *Compositional Controller Synthesis for Stochastic Games*. In Paolo Baldan & Daniele Gorla, editors: *CONCUR 2014 - Concurrency Theory - 25th International Conference, CONCUR 2014, Rome, Italy, September 2-5, 2014. Proceedings, Lecture Notes in Computer Science 8704*, Springer, pp. 173–187, doi:10.1007/978-3-662-44584-6_13.
- [7] Raphaël Berthon, Shibashis Guha & Jean-François Raskin (2020): *Mixing Probabilistic and non-Probabilistic Objectives in Markov Decision Processes*. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi & Dale Miller, editors: *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, ACM, pp. 195–208, doi:10.1145/3373718.3394805.
- [8] Tomáš Brázdil, Václav Brozek, Krishnendu Chatterjee, Vojtech Forejt & Antonín Kucera (2014): *Two Views on Multiple Mean-Payoff Objectives in Markov Decision Processes*. *Log. Methods Comput. Sci.* 10(1), doi:10.2168/LMCS-10(1:13)2014.
- [9] Romain Brenguier & Vojtech Forejt (2016): *Decidability Results for Multi-objective Stochastic Games*. In Cyrille Artho, Axel Legay & Doron Peled, editors: *Automated Technology for Verification and Analysis - 14th International Symposium, ATVA 2016, Chiba, Japan, October 17-20, 2016, Proceedings, Lecture Notes in Computer Science 9938*, pp. 227–243, doi:10.1007/978-3-319-46520-3_15.
- [10] Krishnendu Chatterjee, Luca de Alfaro & Thomas A. Henzinger (2013): *Strategy improvement for concurrent reachability and turn-based stochastic safety games*. *J. Comput. Syst. Sci.* 79(5), pp. 640–657, doi:10.1016/j.jcss.2012.12.001.
- [11] Krishnendu Chatterjee & Laurent Doyen (2016): *Perfect-Information Stochastic Games with Generalized Mean-Payoff Objectives*. In Martin Grohe, Eric Koskinen & Natarajan Shankar, editors: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, ACM, pp. 247–256, doi:10.1145/2933575.2934513.
- [12] Krishnendu Chatterjee, Laurent Doyen & Thomas A. Henzinger (2009): *A Survey of Stochastic Games with Limsup and Liminf Objectives*. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikolettseas & Wolfgang Thomas, editors: *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part II, Lecture Notes in Computer Science 5556*, Springer, pp. 1–15, doi:10.1007/978-3-642-02930-1_1.
- [13] Krishnendu Chatterjee, Laurent Doyen & Thomas A. Henzinger (2013): *A survey of partial-observation stochastic parity games*. *Formal Methods Syst. Des.* 43(2), pp. 268–284, doi:10.1007/s10703-012-0164-2.
- [14] Krishnendu Chatterjee, Vojtech Forejt & Dominik Wojtczak (2013): *Multi-objective Discounted Reward Verification in Graphs and MDPs*. In Kenneth L. McMillan, Aart Middeldorp & Andrei Voronkov, editors: *Logic for Programming, Artificial Intelligence, and Reasoning - 19th International Conference, LPAR-19, Stellenbosch, South Africa, December 14-19, 2013. Proceedings, Lecture Notes in Computer Science 8312*, Springer, pp. 228–242, doi:10.1007/978-3-642-45221-5_17.
- [15] Krishnendu Chatterjee & Thomas A. Henzinger (2008): *Value Iteration*. In Orna Grumberg & Helmut Veith, editors: *25 Years of Model Checking - History, Achievements, Perspectives, Lecture Notes in Computer Science 5000*, Springer, pp. 107–138, doi:10.1007/978-3-540-69850-0_7.
- [16] Krishnendu Chatterjee & Thomas A. Henzinger (2012): *A survey of stochastic ω -regular games*. *J. Comput. Syst. Sci.* 78(2), pp. 394–413, doi:10.1016/j.jcss.2011.05.002.
- [17] Krishnendu Chatterjee, Joost-Pieter Katoen, Maximilian Weininger & Tobias Winkler (2020): *Stochastic Games with Lexicographic Reachability-Safety Objectives*. In Shuvendu K. Lahiri & Chao Wang, editors: *Computer Aided Verification - 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21-24, 2020, Proceedings, Part II, Lecture Notes in Computer Science 12225*, Springer, pp. 398–420, doi:10.1007/978-3-030-53291-8_21.

- [18] Krishnendu Chatterjee, Zuzana Kretínská & Jan Kretínský (2017): *Unifying Two Views on Multiple Mean-Payoff Objectives in Markov Decision Processes*. *Log. Methods Comput. Sci.* 13(2), doi:10.23638/LMCS-13(2:15)2017.
- [19] Taolue Chen, Vojtech Forejt, Marta Z. Kwiatkowska, Aistis Simaitis & Clemens Wiltsche (2013): *On Stochastic Games with Multiple Objectives*. In Krishnendu Chatterjee & Jiri Sgall, editors: *Mathematical Foundations of Computer Science 2013 - 38th International Symposium, MFCS 2013, Klosterneuburg, Austria, August 26-30, 2013. Proceedings, Lecture Notes in Computer Science 8087*, Springer, pp. 266–277, doi:10.1007/978-3-642-40313-2_25.
- [20] Taolue Chen, Marta Z. Kwiatkowska, Aistis Simaitis & Clemens Wiltsche (2013): *Synthesis for Multi-objective Stochastic Games: An Application to Autonomous Urban Driving*. In Kaustubh R. Joshi, Markus Siegle, Mariëlle Stoelinga & Pedro R. D’Argenio, editors: *Quantitative Evaluation of Systems - 10th International Conference, QEST 2013, Buenos Aires, Argentina, August 27-30, 2013. Proceedings, Lecture Notes in Computer Science 8054*, Springer, pp. 322–337, doi:10.1007/978-3-642-40196-1_28.
- [21] Anne Condon (1990): *On Algorithms for Simple Stochastic Games*. In Jin-Yi Cai, editor: *Advances In Computational Complexity Theory, Proceedings of a DIMACS Workshop, New Jersey, USA, December 3-7, 1990, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 13*, DIMACS/AMS, pp. 51–71, doi:10.1090/dimacs/013/04.
- [22] Anne Condon (1992): *The Complexity of Stochastic Games*. *Inf. Comput.* 96(2), pp. 203–224, doi:10.1016/0890-5401(92)90048-K.
- [23] Florent Delgrange, Joost-Pieter Katoen, Tim Quatmann & Mickael Randour (2020): *Simple Strategies in Multi-Objective MDPs*. In Armin Biere & David Parker, editors: *Tools and Algorithms for the Construction and Analysis of Systems - 26th International Conference, TACAS 2020, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2020, Dublin, Ireland, April 25-30, 2020, Proceedings, Part I, Lecture Notes in Computer Science 12078*, Springer, pp. 346–364, doi:10.1007/978-3-030-45190-5_19.
- [24] Kousha Etessami, Marta Z. Kwiatkowska, Moshe Y. Vardi & Mihalis Yannakakis (2008): *Multi-Objective Model Checking of Markov Decision Processes*. *Log. Methods Comput. Sci.* 4(4), doi:10.2168/LMCS-4(4:8)2008.
- [25] Kousha Etessami & Mihalis Yannakakis (2006): *Efficient Qualitative Analysis of Classes of Recursive Markov Decision Processes and Simple Stochastic Games*. In Bruno Durand & Wolfgang Thomas, editors: *STACS 2006, 23rd Annual Symposium on Theoretical Aspects of Computer Science, Marseille, France, February 23-25, 2006, Proceedings, Lecture Notes in Computer Science 3884*, Springer, pp. 634–645, doi:10.1007/11672142_52.
- [26] Lu Feng, Clemens Wiltsche, Laura R. Humphrey & Ufuk Topcu (2015): *Controller synthesis for autonomous systems interacting with human operators*. In Alexandre M. Bayen & Michael S. Branicky, editors: *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems, ICCPS 2015, Seattle, WA, USA, April 14-16, 2015*, ACM, pp. 70–79, doi:10.1145/2735960.2735973.
- [27] Nathanaël Fijalkow & Florian Horn (2010): *The surprising complexity of reachability games*. *CoRR* abs/1010.2420. Available at <http://arxiv.org/abs/1010.2420>.
- [28] Jerzy A Filar, Dmitry Krass & Keith W Ross (1995): *Percentile performance criteria for limiting average Markov decision processes*. *IEEE Transactions on Automatic Control* 40(1), pp. 2–10, doi:10.1109/9.362904.
- [29] Vojtech Forejt, Marta Z. Kwiatkowska, Gethin Norman, David Parker & Hongyang Qu (2011): *Quantitative Multi-objective Verification for Probabilistic Systems*. In Parosh Aziz Abdulla & K. Rustan M. Leino, editors: *Tools and Algorithms for the Construction and Analysis of Systems - 17th International Conference, TACAS 2011, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2011, Saarbrücken, Germany, March 26-April 3, 2011. Proceedings, Lecture Notes in Computer Science 6605*, Springer, pp. 112–127, doi:10.1007/978-3-642-19835-9_11.
- [30] Jan Kretínský, Emanuel Ramneantu, Alexander Slivinskiy & Maximilian Weininger (2020): *Comparison of Algorithms for Simple Stochastic Games*. In Jean-François Raskin & Davide Bresolin, editors: *Proceed-*

- ings 11th International Symposium on Games, Automata, Logics, and Formal Verification, GandALF 2020, Brussels, Belgium, September 21-22, 2020, EPTCS 326, pp. 131–148, doi:10.4204/EPTCS.326.9.
- [31] Marta Kwiatkowska, Gethin Norman, David Parker & Gabriel Santos (2020): *Multi-player Equilibria Verification for Concurrent Stochastic Games*. In Marco Gribaudo, David N. Jansen & Anne Remke, editors: *Quantitative Evaluation of Systems - 17th International Conference, QEST 2020, Vienna, Austria, August 31 - September 3, 2020, Proceedings, Lecture Notes in Computer Science 12289*, Springer, pp. 74–95, doi:10.1007/978-3-030-59854-9_7.
- [32] Marta Kwiatkowska, Gethin Norman, David Parker & Gabriel Santos (2020): *PRISM-games 3.0: Stochastic Game Verification with Concurrency, Equilibria and Time*. In Shuvendu K. Lahiri & Chao Wang, editors: *Computer Aided Verification - 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21-24, 2020, Proceedings, Part II, Lecture Notes in Computer Science 12225*, Springer, pp. 475–487, doi:10.1007/978-3-030-53291-8_25.
- [33] Marta Kwiatkowska, David Parker & Clemens Wiltsche (2018): *PRISM-games: verification and strategy synthesis for stochastic multi-player games with multiple objectives*. *Int. J. Softw. Tools Technol. Transf.* 20(2), pp. 195–210, doi:10.1007/s10009-017-0476-z.
- [34] Kim G. Larsen, Marius Mikucionis, Marco Muñoz, Jiri Srba & Jakob Haahr Taankvist (2016): *Online and Compositional Learning of Controllers with Application to Floor Heating*. In Marsha Chechik & Jean-François Raskin, editors: *Tools and Algorithms for the Construction and Analysis of Systems - 22nd International Conference, TACAS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings, Lecture Notes in Computer Science 9636*, Springer, pp. 244–259, doi:10.1007/978-3-662-49674-9_14.
- [35] Mickael Randour, Jean-François Raskin & Ocan Sankur (2017): *Percentile queries in multi-dimensional Markov decision processes*. *Formal Methods Syst. Des.* 50(2-3), pp. 207–248, doi:10.1007/s10703-016-0262-7.
- [36] Lloyd S Shapley (1953): *Stochastic games*. *Proceedings of the national academy of sciences* 39(10), pp. 1095–1100, doi:10.1073/pnas.39.10.1095.
- [37] Yaron Verner (2015): *Robust Multidimensional Mean-Payoff Games are Undecidable*. In Andrew M. Pitts, editor: *Foundations of Software Science and Computation Structures - 18th International Conference, FoS-SaCS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings, Lecture Notes in Computer Science 9034*, Springer, pp. 312–327, doi:10.1007/978-3-662-46678-0_20.
- [38] Tobias Winkler & Maximilian Weininger (2021): *Stochastic Games with Disjunctions of Multiple Objectives (Technical Report)*. CoRR abs/2108.04604. Available at <https://arxiv.org/abs/2108.04604>.