On Stochastic Games with Multiple Objectives

Taolue Chen, Vojtěch Forejt, Marta Kwiatkowska,
Aistis Simaitis, and Clemens Wiltsche

Department of Computer Science, University of Oxford, United Kingdom

Abstract. We study two-player stochastic games, where the goal of one player is to satisfy a formula given as a positive boolean combination of expected total reward objectives and the behaviour of the second player is adversarial. Such games are important for modelling, synthesis and verification of open systems with stochastic behaviour. We show that finding a winning strategy is PSPACE-hard in general and undecidable for deterministic strategies. We also prove that optimal strategies, if they exists, may require infinite memory and randomisation. However, when restricted to disjunctions of objectives only, memoryless deterministic strategies suffice, and the problem of deciding whether a winning strategy exists is NP-complete. We also present algorithms to approximate the Pareto sets of achievable objectives for the class of stopping games.

1 Introduction

Stochastic games [20] have many applications in semantics and formal verification, and have been used as abstractions for probabilistic systems [15], and more recently for quantitative verification and synthesis of competitive stochastic systems [8]. Two-player games, in particular, provide a natural representation of open systems, where one player represents the system and the other its environment, in this paper referred to as Player 1 and Player 2, respectively. Stochasticity models uncertainty or randomisation, and leads to a game where each player can select an outgoing edge in states he controls, while in stochastic states the choice is made according to a state-dependent probability distribution. A *strategy* describes which actions a player picks. A fixed pair of strategies and an initial state determines a probability space on the runs of a game, and yields expected values of given objective (payoff) functions. The problem is then to determine if Player 1 has a strategy to ensure that the expected values of the objective functions meet a given set of criteria for all strategies that Player 2 may choose.

Various objective functions have been studied, for example reachability, ω-regular, or parity [4]. We focus here on *reward functions*, which are determined by a reward structure, annotating states with rewards. A prominent example is the reward function evaluating *total reward*, which is obtained by summing up rewards for all states visited along a path. Total rewards can be conveniently used to model consumption of resources along the execution of the system, but (with a straightforward modification of the game) they can also be used to encode other objective functions, such as reachability.

Although objective functions can express various useful properties, many situations demand considering not just the value of a single objective function, but rather values of several such functions simultaneously. For example, we may wish to maximise the number of successfully provided services and, at the same time, ensure minimising resource usage. More generally, given multiple objective functions, one may ask whether an arbitrary boolean combination of upper or lower bounds on the expected values of these functions can be ensured (in this paper we restrict only to positive boolean combinations, i.e. we do not allow negations). Alternatively, one might ask to compute or approximate the *Pareto set*, i.e. the set of all bounds that can be assured by exploring trade-offs. The simultaneous optimisation of a conjunction of objectives (also known as multi-objective, multi-criteria or multi-dimensional optimisation) is actively studied in operations research [21] and used in engineering [17]. In verification it has been considered for Markov decision processes (MDPs), which can be seen as one-player stochastic games, for discounted objectives [5] and general ω-regular objectives [10]. Multiple objectives for non-stochastic games have been studied by a number of authors, including in the context of energy games [22] and strategy synthesis [6].

In this paper, we study *stochastic games* with multi-objective queries, which are expressed as positive boolean combinations of total reward functions with upper or lower bounds on the expected reward to be achieved. In that way we can, for example, give several alternatives for a valid system behaviour, such as "the expected consumption of the system is at most 10 units of energy and the probability of successfully finishing the operation is at least 70%, or the expected consumption is at most 50 units, but the probability of success is at least 99%". Another motivation for our work is assume-guarantee compositional verification [19], where the system satisfies a set of guarantees φ whenever a set of assumptions ψ is true. This can be formulated using multi-objective queries of the form $\bigwedge\psi \Rightarrow \bigwedge\varphi$. For MDPs it has been shown how to formulate assume-guarantee rules using multi-objective queries [10]. The results obtained in this paper would enable us to explore the extension to stochastic games.

Contributions. We first obtain nondeterminacy by a straightforward modification of earlier results. Then we prove the following novel results for multi-objective stochastic games:

- We prove that, even in a pure conjunction of objectives, infinite memory and randomisation are required for the winning strategy of Player 1, and that the problem of finding a *deterministic* winning strategy is undecidable.
- For the case of a pure disjunction of objectives, we show that memoryless deterministic strategies are sufficient for Player 1 to win, and we prove that determining the existence of such strategies is an NP-complete problem.
- For the general case, we show that the problem of deciding whether Player 1 has a winning strategy in a game is PSPACE-hard.
- We provide Pareto set approximation algorithms for stopping games. This result directly applies to the important class of *discounted rewards* for non-stopping games, due to an off-the-shelf reduction [9].

Related work. Multi-objective optimisation has been studied for various subclasses of stochastic games. For non-stochastic games, multi-dimensional objectives have been considered in [6,22]. For MDPs, multiple discounted objectives [5], long-run objectives [2], ω-regular objectives [10] and total rewards [12] have been analysed. The objectives that we study in this paper are a special case of branching time temporal logics for stochastic games [3,1]. However, already for MDPs, such logics are so powerful that it is not decidable whether there is an optimal controller [3]. A special case of the problem studied in this paper is the case where the goal of Player 1 is to achieve a *precise value* of the expectation of an objective function [9]. As regards applications, stochastic games with a single objective function have been employed and implemented for quantitative abstraction refinement for MDP models in [15]. The usefulness of techniques for verification and strategy synthesis for stochastic games with a single objective is demonstrated, e.g., for smart grid protocols [8]. Applications of multi-objective verification include assume-guarantee verification [16] and controller synthesis [13] for MDPs.

2 Preliminaries

We begin this section by introducing notations used throughout the paper. We then provide the definition of stochastic two-player games together with the concepts of strategies and paths of the game. Finally, we introduce the objectives that are studied in this paper.

2.1 Notation

Given a vector $\boldsymbol{x} \in \mathbb{R}^n$, we use x_i to refer to its i-th component, where $1 \leq i \leq n$, and define the norm $\|\boldsymbol{x}\| \stackrel{\text{def}}{=} \sum_{i=1}^{n} |x_i|$. Given a number $y \in \mathbb{R}$, we use $\boldsymbol{x} \pm y$ to denote the vector $(x_1 \pm y, x_2 \pm y, \ldots, x_n \pm y)$. Given two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, the *dot product* of $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined by $\boldsymbol{x} \cdot \boldsymbol{y} = \sum_{i=1}^{n} x_i \cdot y_i$, and the comparison operator $\leq$ on vectors is defined to be the componentwise ordering. The sum of two sets of vectors $X, Y \subseteq \mathbb{R}^n$ is defined by $X + Y = \{\boldsymbol{x} + \boldsymbol{y} \,|\, \boldsymbol{x} \in X, \boldsymbol{y} \in Y\}$. Given a set X, we define the *downward closure* of X as $\mathsf{dwc}(X) \stackrel{\text{def}}{=} \{\boldsymbol{y} \mid \exists \boldsymbol{x} \in X \,.\, \boldsymbol{y} \leq \boldsymbol{x}\}$ and the *upward closure* as $\mathsf{up}(X) \stackrel{\text{def}}{=} \{\boldsymbol{y} \,|\, \exists \boldsymbol{x} \in X \,.\, \boldsymbol{x} \leq \boldsymbol{y}\}$. We denote by $\mathbb{R}_{\pm\infty}$ the set $\mathbb{R} \cup \{+\infty, -\infty\}$, and we define the operations $\cdot$ and $+$ in the expected way, defining $0 \cdot x = 0$ for all $x \in \mathbb{R}_{\pm\infty}$ and leaving $-\infty + \infty$ undefined. We also define function $\mathrm{sgn}(x) : \mathbb{R}_{\pm\infty} \to \mathbb{N}$ to be 1 if $x > 0$, -1 if $x < 0$ and 0 if $x = 0$.

A *discrete probability distribution* (or just *distribution*) over a (countable) set S is a function $\mu : S \to [0, 1]$ such that $\sum_{s \in S} \mu(s) = 1$. We write $\mathcal{D}(S)$ for the set of all distributions over S. Let $\mathsf{supp}(\mu) = \{s \in S \mid \mu(s) > 0\}$ be the *support set* of $\mu \in \mathcal{D}(S)$. We say that a distribution $\mu \in \mathcal{D}(S)$ is a *Dirac distribution* if $\mu(s) = 1$ for some $s \in S$. We represent a distribution $\mu \in \mathcal{D}(S)$ on a set $S = \{s_1, \ldots, s_n\}$ as a map $[s_1 \mapsto \mu(s_1), \ldots, s_n \mapsto \mu(s_n)]$ and omit the elements of S outside $\mathsf{supp}(\mu)$ to simplify the presentation. If the context is clear we sometimes identify a Dirac distribution μ with the unique element in $\mathsf{supp}(\mu)$.

2.2 Stochastic games

In this section we introduce turn-based stochastic two-player games.

Stochastic two-player games. A *stochastic two-player game* is a tuple $\mathcal{G} = \langle S, (S_\Box, S_\Diamond, S_\bigcirc), \Delta \rangle$ where S is a finite set of states partitioned into sets $S_\Box$, $S_\Diamond$, and $S_\bigcirc$; $\Delta : S \times S \to [0,1]$ is a probabilistic transition function such that $\Delta(\langle s,t \rangle) \in \{0,1\}$ if $s \in S_\Box \cup S_\Diamond$ and $\sum_{t \in S} \Delta(\langle s,t \rangle) = 1$ if $s \in S_\bigcirc$.

$S_\Box$ and $S_\Diamond$ represent the sets of states controlled by players Player 1 and Player 2, respectively, while $S_\bigcirc$ is the set of stochastic states. For a state $s \in S$, the set of successor states is denoted by $\Delta(s) \stackrel{\text{def}}{=} \{t \in S \mid \Delta(\langle s,t \rangle) > 0\}$. We assume that $\Delta(s) \neq \emptyset$ for all $s \in S$. A state from which no other states except for itself are reachable is called *terminal*, and the set of terminal states is denoted by $\mathsf{Term} \stackrel{\text{def}}{=} \{s \in S \mid \Delta(\langle s,t \rangle) = 1 \text{ iff } s = t\}$.

Paths. An infinite *path* λ of a stochastic game $\mathcal{G}$ is an infinite sequence $s_0 s_1 \ldots$ of states such that $s_{i+1} \in \Delta(s_i)$ for all $i \geq 0$. A finite path is a finite such sequence. For a finite or infinite path λ we write $\mathsf{len}(\lambda)$ for the number of states in the path. For $i < \mathsf{len}(\lambda)$ we write λ_i to refer to the i-th state s_i of λ. For a finite path λ we write $\mathsf{last}(\lambda)$ for the last state of the path. For a game $\mathcal{G}$ we write $\Omega^+_\mathcal{G}$ for the set of all finite paths, and $\Omega_\mathcal{G}$ for the set of all infinite paths, and $\Omega_{\mathcal{G},s}$ for the set of infinite paths starting in state s. We denote the set of paths that reach a state in $T \subseteq S$ by $\Diamond T \stackrel{\text{def}}{=} \{\omega \in \Omega_\mathcal{G} \mid \exists i \,.\, \omega_i \in T\}$.

Strategies. A *strategy* of Player 1 is a (partial) function $\pi : \Omega^+_\mathcal{G} \to \mathcal{D}(S)$, which is defined for $\lambda \in \Omega^+_\mathcal{G}$ only if $\mathsf{last}(\lambda) \in S_\Box$, such that $s \in \mathsf{supp}(\pi(\lambda))$ only if $\Delta(\langle \mathsf{last}(\lambda), s \rangle) = 1$. A strategy π is a *finite-memory* strategy if there is a finite automaton $\mathcal{A}$ over the alphabet S such that $\pi(\lambda)$ is determined by $\mathsf{last}(\lambda)$ and the state of $\mathcal{A}$ in which it ends after reading the word λ. We say that π is *memoryless* if $\mathsf{last}(\lambda) = \mathsf{last}(\lambda')$ implies $\pi(\lambda) = \pi(\lambda')$, and *deterministic* if $\pi(\lambda)$ is Dirac for all $\lambda \in \Omega^+_\mathcal{G}$. If π is a memoryless strategy for Player 1 then we identify it with the mapping $\pi : S_\Box \to \mathcal{D}(S)$. A strategy σ for Player 2 is defined similarly. We denote by Π and Σ the sets of all strategies for Player 1 and Player 2, respectively.

Probability measures. A stochastic game $\mathcal{G}$, together with a strategy pair $(\pi, \sigma) \in \Pi \times \Sigma$ and a starting state s, induces an infinite Markov chain on the game (see e.g. [9]). We define the probability measure of this Markov chain by $\Pr^{\pi,\sigma}_{\mathcal{G},s}$. The expected value of a measurable function $f : S^\omega \to \mathbb{R}_{\pm\infty}$ is defined as $\mathbb{E}^{\pi,\sigma}_{\mathcal{G},s}[f] \stackrel{\text{def}}{=} \int_{\Omega_{\mathcal{G},s}} f \, d\Pr^{\pi,\sigma}_{\mathcal{G},s}$. We say that a game $\mathcal{G}$ is a *stopping game* if, for every pair of strategies π and σ, a terminal state is reached with probability 1.

Rewards. A reward function $\boldsymbol{r} : S \to \mathbb{Q}^n$ assigns a reward vector $\boldsymbol{r}(s) \in \mathbb{Q}^n$ to each state s of the game $\mathcal{G}$. We use r_i for the function defined by $r_i(t) = \boldsymbol{r}(t)_i$ for all t. We assume that for each i the reward assigned by r_i is either non-negative or non-positive for all states (we adopt this approach in order to express minimisation problems via maximisation, as explained in the next subsection). The analysis of more general reward functions is left for future work. We define

the vector of *total reward* random variables $rew(\boldsymbol{r})$ such that, given a path λ, $rew(\boldsymbol{r})(\lambda) = \sum_{j \geq 0} \boldsymbol{r}(\lambda_j)$.

2.3 Multi-objective queries

A *multi-objective query* (MQ) φ is a positive boolean combination (i.e. disjunctions and conjunctions) of predicates (or *objectives*) of the form $r \bowtie v$, where r is a reward function, $v \in \mathbb{Q}$ is a bound and $\bowtie \in \{\geq, \leq\}$ is a comparison operator. The validity of an MQ is defined inductively on the structure of the query: an objective $r \bowtie v$ is true in a state s of $\mathcal{G}$ under a pair of strategies (π, σ) if and only if $\mathbb{E}^{\pi,\sigma}_{\mathcal{G},s}[rew(r)] \bowtie v$, and the truth value of disjunctions and conjunctions of queries is defined straightforwardly. Using the definition of the reward function above, we can express the operator $\leq$ by using $\geq$, applying the equivalence $r \leq v \equiv (-r \geq -v)$. Thus, throughout the paper we often assume that MQs only contain the operator $\geq$.

We say that Player 1 *achieves* the MQ φ (i.e., *wins* the game) in a state s if it has a strategy π such that for all strategies σ of Player 2 the query φ evaluates to true under (π, σ). An MQ φ is a *conjunctive query* (CQ) if it is a conjunction of objectives, and a *disjunctive query* (DQ) if it is a disjunction of objectives.

For a MQ φ containing n objectives $r_i \bowtie_i v_i$ for $1 \leq i \leq n$ and for $\boldsymbol{x} \in \mathbb{R}^n$ we use $\varphi[\boldsymbol{x}]$ to denote φ in which each $r_i \bowtie_i v_i$ is replaced with $r_i \bowtie_i x_i$.

Reachability. We can enrich multi-objective queries with *reachability objectives*, i.e. objectives $\Diamond T \geq p$ for a set of target states $T \subseteq S$, where $p \in [0, 1]$ is a bound. The objective $\Diamond T \geq p$ is true under a pair of strategies (π, σ) if $\Pr^{\pi,\sigma}_{\mathcal{G},s}(\Diamond T) \geq p$, and notions such as achieving a query are defined straightforwardly. Note that queries containing reachability objectives can be reduced to queries with total expected reward only (see [7] for a reduction). It also follows from the construction that if all target sets contain only terminal states, the reduction works in polynomial time.

Pareto sets. Let φ be an MQ containing n objectives. The vector $\boldsymbol{v} \in \mathbb{R}^n$ is a *Pareto vector* if and only if (a) $\varphi[\boldsymbol{v} - \varepsilon]$ is achievable for all $\varepsilon > 0$, and (b) $\varphi[\boldsymbol{v} + \varepsilon]$ is not achievable for any $\varepsilon > 0$. The set P of all such vectors is called a *Pareto set*. Given $\varepsilon > 0$, an *ε-approximation of a Pareto set* is a set of vectors Q satisfying that, for any $\boldsymbol{w} \in Q$, there is a vector $\boldsymbol{v}$ in the Pareto set such that $\|\boldsymbol{v} - \boldsymbol{w}\| \leq \varepsilon$, and for every $\boldsymbol{v}$ in the Pareto set there is a vector $\boldsymbol{w} \in Q$ such that $\|\boldsymbol{v} - \boldsymbol{w}\| \leq \varepsilon$.

Example. Consider the game $\mathcal{G}$ from Figure 1 (left). It consists of one Player 1 state s_0, one Player 2 state s_1, six stochastic states s_2, s_3, s_4, s_5, t_1 and t_2, as well as two terminal states t'_1 and t'_2. Outgoing edges of stochastic states are assigned uniform distributions by convention. For the MQ $\varphi_1 = r_1 \geq \frac{2}{3} \wedge r_2 \geq \frac{1}{6}$, where the reward functions are defined by $r_1(t_1){=}r_2(t_2){=}1$ and all other values are zero, the Pareto set for the initial state s_0 is shown in Figure 1 (centre). Hence, φ_1 is satisfied at s_0, as $(\frac{2}{3}, \frac{1}{6})$ is in the Pareto set. For the MQ $\varphi_2 = r_1 \geq \frac{2}{3} \wedge -r_2 \geq -\frac{1}{6}$, Figure 1 (right) illustrates the Pareto set for s_0, showing that φ_2 is not satisfied

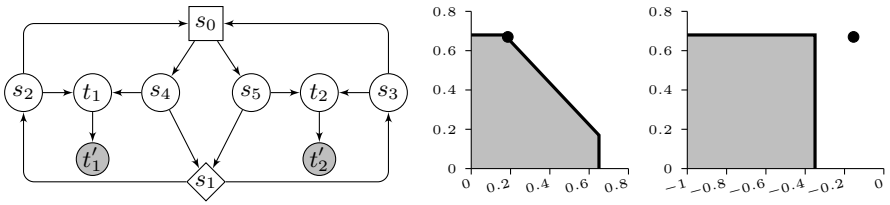

Fig. 1: An example game (left), Pareto set for φ_1 at s_0 (centre), and Pareto set for φ_2 at s_0 (right), with bounds indicated by a dot. Note that the sets are unbounded towards $-\infty$.

at s_0. Note that φ_1 and φ_2 correspond to the combination of reachability and safety objectives, i.e., $\Diamond\{t'_1\} \geq \frac{2}{3} \wedge \Diamond\{t'_2\} \geq \frac{1}{6}$ and $\Diamond\{t'_1\} \geq \frac{2}{3} \wedge \Diamond\{t'_2\} \leq \frac{1}{6}$.

3 Conjunctions of Objectives

In this section we present the results for CQs. We first recall that the games are not determined, and then show that Player 1 may require an infinite-memory randomised strategy to win, while it is not decidable whether deterministic winning strategies exist. We also provide fixpoint equations characterising the Pareto sets of achievable vectors and their successive approximations.

Theorem 1 (Non-determinacy, optimal strategies [9]). *Stochastic games with multiple objectives are, in general, not determined, and optimal strategies might not exist, already for CQs with two objectives.*

Theorem 1 carries over from the results for precise value games, because the problem of reaching a set of terminal states $T \subseteq \mathsf{Term}$ with probability precisely p is a special case of multi-objective stochastic games and can be expressed as a CQ $\varphi = \Diamond T \geq p \wedge \Diamond T \leq p$.

Theorem 2 (Infinite memory). *An infinite-memory randomised strategy may be required for Player 1 to win a multi-objective stochastic game with a CQ even for stopping games with reachability objectives.*

Proof. To prove the theorem we will use the example game from Figure 2. We only explain the intuition behind the need of infinite memory here; the formal proof is presented in [7]. First, we note that it is sufficient to consider deterministic counter-strategies for Player 2, since, after Player 1 has proposed his strategy, the resulting model is an MDP with finite branching [18]. Consider the game starting in the initial state s_0 and a CQ $\varphi = \bigwedge_{i=1}^{3} \Diamond T_i \geq \frac{1}{3}$, where the target sets T_1, T_2 and T_3 contain states labelled 1, 2 and 3, respectively. We note that target sets are terminal and disjoint, and for any π and σ we have that $\sum_{i=1}^{3} \Pr^{\pi,\sigma}_{\mathcal{G},s_0}(\Diamond T_i) = 1$, and hence for any winning Player 1 strategy π it must be the case that, for any σ, $\Pr^{\pi,\sigma}_{\mathcal{G},s_0}(\Diamond T_i) = \frac{1}{3}$ for $1 \leq i \leq 3$.

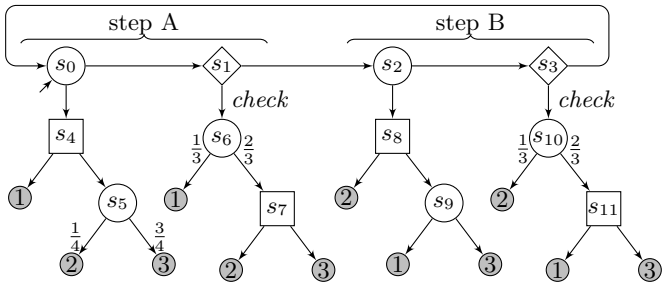

Fig. 2: Game where Player 1 requires infinite memory to win.

Let E be the set of runs which never take any transition *check*. The game proceeds by alternating between the two steps A and B as indicated in Figure 2. In step A, Player 1 chooses a probability to go to T_1 from state s_4, and then Player 2 gets an opportunity to "verify" that the probability $\Pr^{\pi,\sigma}_{\mathcal{G},s_0}(\Diamond T_1|E)$ of runs reaching T_1 conditional on the event that no *check* action was taken is $\frac{1}{3}$. She can do this by taking the action *check* and so ensuring that $\Pr^{\pi,\sigma}_{\mathcal{G},s_0}(\Diamond T_1|\Omega_{\mathcal{G}} \setminus E) = \frac{1}{3}$. If Player 2 again does not choose to take *check*, the game continues in step B, where the same happens for T_2, and so on.

When first performing step A, Player 1 has to pick probability $\frac{1}{3}$ to go to T_1. But since the probability of going from s_4 to T_2 is $< \frac{1}{3}$, when step B is performed for the first time, Player 1 must go to T_2 with probability $y_0 > \frac{1}{3}$ to compensate for the "loss" of the probability in step A. However, this decreases the probability of reaching T_1 at step B, and so Player 1 must compensate for it in the subsequent step A by taking probability $> \frac{1}{3}$ of going to T_1. This decreases the probability of reaching T_2 in the second step B even more (compared to first execution of step A), for which Player 1 must compensate by picking $y_1 > y_0 > \frac{1}{3}$ in the second execution of step B, and so on. So, in order to win, Player 1 has to play infinitely many different probability distributions in states s_4 and s_8. Note that, if Player 2 takes action "check", Player 1 can always randomise in states s_7 and s_{11} to achieve expectations exactly $\frac{1}{3}$ for all objectives. □

In fact, the above idea allows us to encode natural numbers together with operations of increment and decrement, and obtain a reduction of the location reachability problem in the two-counter machine (which is known to be undecidable [14]) to the problem of deciding whether there exists a *deterministic* winning strategy for Player 1 in a multi-objective stochastic game.

Theorem 3 (Undecidability). *The problem whether there exists a* deterministic *winning strategy for Player 1 in a multi-objective stochastic game is undecidable already for stopping games and conjunctions of reachability objectives.*

Our proof is inspired by the proof of [3] which shows that the problem of existence of a winning strategy in an MDP for a PCTL formula is undecidable. However, the proof of [3] relies on branching time features of PCTL to ensure the counter

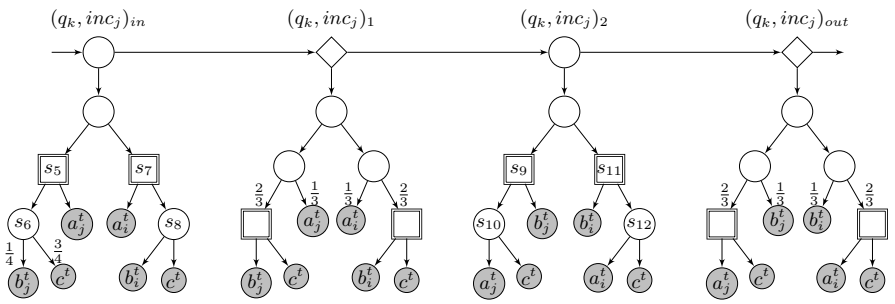

Fig. 3: Increment gadget for counter j.

values of the two-counter machine are encoded correctly. Since MQs only allow us to express combinations of linear-time properties, we need to take a different approach, utilising ideas of Theorem 2. We present the proof idea here; for the full proof see [7]. We encode the counter machine instructions in gadgets similar to the ones used for the proof of Theorem 2, where Player 1 has to change the probabilities with which he goes to the target states based on the current value of the counter. For example, the gadget in Figure 3 encodes the instruction to *increment* the counter j. The basic idea is that, if the counter value is c_j when entering the increment gadget, then in state s_5 Player 1 has to assign probability exactly $\frac{2}{3 \cdot 2^{c_j}}$ to the edge $\langle s_5, s_6 \rangle$, and then probability $\frac{2}{3 \cdot 2^{c_j+1}}$ to the edge $\langle s_9, s_{10} \rangle$ in s_9, resulting in the counter being incremented. The gadgets for counter decrement and zero-check can be found in [7]. The resulting query contains six target sets. In particular, there is a conjunct $\Diamond T_t \geq 1$, where the set T_t is not reached with probability 1 only if the gadget representing the target counter machine location is reached. The remaining five objectives ensure that Player 1 updates the counter values correctly (by picking corresponding probability distributions) and so the strategy encodes a valid computation of the two-counter machine. Hence, the counter machine terminates if and only if there does not exist a winning strategy for Player 1.

We note that the problem of deciding whether there is a *randomised* winning strategy for Player 1 remains open, since the gadgets modelling decrement instructions in our construction rely on the strategy being deterministic. Nevertheless, for stopping games, in Theorem 4 below we provide a functional that, given a CQ φ, computes $\varepsilon-$approximations of the Pareto sets, i.e. the sets containing the bounds $\boldsymbol{x}$ so that Player 1 has a winning strategy for $\varphi[\boldsymbol{x} - \varepsilon]$. As a corollary of the theorem, using a simple reduction (see e.g. [9]) we get an approximation algorithm for the Pareto sets in non-stopping games with (multiple) *discounted reward objectives*.

Theorem 4 (Pareto set approximation). *For a stopping game $\mathcal{G}$ and a CQ $\varphi = \bigwedge_{i=1}^{n} r_i \geq v_i$, an $\varepsilon-$approximation of the Pareto sets for all states can be computed in $k = |S| + \lceil |S| \cdot \frac{\ln(\varepsilon \cdot (n \cdot M)^{-1})}{\ln(1-\delta)} \rceil$ iterations of the operator $F : (S \to$*

$\mathcal{P}(\mathbb{R}^n)) \to (S \to \mathcal{P}(\mathbb{R}^n))$ *defined by*

$$F(X)(s) \stackrel{\text{def}}{=} \begin{cases} \mathit{dwc}(\mathit{conv}(\bigcup_{t\in\Delta(s)} X_t)+\boldsymbol{r}(s)) & \text{if } s \in S_\Box \\ \mathit{dwc}(\bigcap_{t\in\Delta(s)} X_t+\boldsymbol{r}(s)) & \text{if } s \in S_\Diamond \\ \mathit{dwc}(\sum_{t\in\Delta(s)} \Delta(\langle s,t\rangle)\cdot X_t+\boldsymbol{r}(s)) & \text{if } s \in S_\bigcirc, \end{cases}$$

where the initial sets are $X_s^0 \stackrel{\text{def}}{=} \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} \leq \boldsymbol{r}(s)\}$ *for all* $s \in S$*, and* $M = |S| \cdot \frac{\max_{s\in S,i} |r_i(s)|}{\delta}$ *for* $\delta = p_{\min}^{|S|}$ *and* $p_{\min}$ *being the smallest positive probability in* $\mathcal{G}$*.*

We first explain the intuition behind the operations when $\boldsymbol{r}(s) = \mathbf{0}$. For $s \in S_\Box$, Player 1 can randomise between successor states, so any convex combination of achievable points in X_t^{k-1} for the successors $t \in \Delta(s)$ is achievable in X_s^k, and so we take the convex closure of the union. For $s \in S_\Diamond$, a value in X_s^k is achievable if it is achievable in X_s^{k-1} for all successors $t \in \Delta(s)$, and hence we take the intersection. Finally, stochastic states $s \in S_\bigcirc$ are like Player 1 states with a fixed probability distribution, and hence the operation performed is the weighted Minkowski sum. When $\boldsymbol{r}(s) \neq \mathbf{0}$, the reward is added as a contribution to what is achievable at s.

Proof (Outline). The proof, presented in [7], consists of two parts. First, we prove that the result of the k-th iteration of F contains exactly the points achievable by some strategy in k steps; this is done by applying induction on k. As the next step, we observe that, since the game is *stopping*, after $|S|$ steps the game has terminated with probability at least $\delta = p_{\min}^{|S|}$. Hence, the maximum change to any dimension to any vector in X_s^k after k steps of the iteration is less than $M \cdot (1-\delta)^{\lfloor \frac{k}{|S|} \rfloor}$. It follows that $k = |S| + \lceil |S| \cdot \frac{\ln(\varepsilon\cdot(n\cdot M)^{-1})}{\ln(1-\delta)} \rceil$ iterations of F suffice to yield all points which are within ε from the Pareto points for $\boldsymbol{r}$.

4 General Multi-Objective Queries

In this section we consider the general case where the objective is expressed as an arbitrary MQ. The nondeterminacy result from Theorem 1 carries over to the more general MQs, and, even if we restrict to DQs, the games stay nondetermined (see [7] for a proof). The following theorem establishes lower complexity bounds for the problem of deciding the existence of the winning strategy for Player 1.

Theorem 5. *The problem of deciding whether there is a winning strategy for Player 1 for an MQ* φ *is PSPACE-hard in general, and NP-hard if* φ *is a DQ.*

The above theorem is proved by reductions from QBF and 3SAT, respectively (see [7] for the proofs). The reduction from QBF is similar to the one in [11], the major differences being that our results apply even when the target states are terminal, and that we need to deal with possible randomisation of the strategies.

We now establish conditions under which a winning strategy for Player 1 exists. Before we proceed, we note that it suffices to consider MQs in conjunctive

normal form (CNF) that contain no negations, since any MQ can be converted to CNF using standard methods of propositional logic. Before presenting the proof of Theorem 6, we give the following reformulation of the separating hyperplane theorem, proved in [7].

Lemma 1. *Let $W \subseteq \mathbb{R}^m_{\pm\infty}$ be a convex set satisfying the following. For all j, whenever there is $\boldsymbol{x} \in W$ such that $sgn(x_j) \geq 0$ (resp. $sgn(x_j) \leq 0$), then $sgn(y_j) \geq 0$ (resp. $sgn(y_j) \leq 0$) for all $\boldsymbol{y} \in W$. Let $\boldsymbol{z} \in \mathbb{R}^m$ be a point which does not lie in the closure of $\mathsf{up}(W)$. Then there is a non-zero vector $\boldsymbol{x} \in \mathbb{R}^m$ such that the following conditions hold:*

1. *for all $1 \leq j \leq m$ we have $x_j \geq 0$;*
2. *for all $1 \leq j \leq m$, if there is $\boldsymbol{w} \in W$ satisfying $w_j = -\infty$, then $x_j = 0$; and*
3. *for all $\boldsymbol{w} \in W$, the product $\boldsymbol{w} \cdot \boldsymbol{x}$ is defined and satisfies $\boldsymbol{w} \cdot \boldsymbol{x} \geq \boldsymbol{z} \cdot \boldsymbol{x}$.*

Theorem 6. *Let $\psi = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{m} q_{i,j} \geq u_{i,j}$ be an MQ in CNF, and let π be a strategy of Player 1. The following two conditions are equivalent.*

- *The strategy π achieves ψ.*
- *For all $\varepsilon > 0$ there are nonzero vectors $\boldsymbol{x}_1, \ldots \boldsymbol{x}_n \in \mathbb{R}^m_{\geq 0}$, such that π achieves the conjunctive query $\varphi = \bigwedge_{i=1}^{n} r_i \geq v_i$, where $r_i(s) = \boldsymbol{x}_i \cdot (q_{i,1}(s), \ldots, q_{i,m}(s))$ and $v_i = \boldsymbol{x}_i \cdot (u_{i,1} - \varepsilon, \ldots, u_{i,m} - \varepsilon)$ for all $1 \leq i \leq n$.*

Proof (Sketch). We only present high-level intuition here, see [7] for the full proof. Using the separating hyperplane theorem we show that if there exists a winning strategy for Player 1, then there exist separating hyperplanes, one per conjunct, separating the objective vectors within each conjunct from the set of points that Player 2 can enforce, and vice versa. This allows us to reduce the MQ expressed in CNF into a CQ, by obtaining one reward function per conjuct, which is constructed by weighthing the original reward function by the characteristic vector of the hyperplane.

When we restrict to DQs only, it follows from Theorem 6 that there exists a strategy achieving a DQ if and only if there is a strategy achieving a certain single-objective expected total reward, and hence we obtain the following theorem.

Theorem 7 (Memoryless deterministic strategies). *Memoryless deterministic strategies are sufficient for Player 1 to achieve a DQ.*

Since memoryless deterministic strategies suffice for optimising single total reward, to determine whether a DQ is achievable we can guess such a strategy for Player 1, which uniquely determines an MDP. We can then use the polynomial time algorithm of [10] to verify that there exists no winning Player 2 strategy. This NP algorithm, together with Theorem 5, gives us the following corollary.

Corollary 1. *The problem whether a DQ is achievable is NP-complete.*

Using Theorem 6 we can construct an approximation algorithm computing Pareto sets for disjunctive objectives for stopping games, which performs multiple calls to the algorithm for computing optimal value for the single-objective reward.

Theorem 8 (Pareto sets). *For stopping games, given a vector* $\boldsymbol{r} = (r_1, \ldots, r_m)$ *of reward functions, an* ε*-approximation of the Pareto sets for disjunction of objectives for* $\boldsymbol{r}$ *can be computed by* $(\frac{2 \cdot m^2 \cdot (M+1)}{\varepsilon})^{m-1}$ *calls to a NP∩coNP algorithm computing single-objective total reward, where* M *is as in Theorem 4.*

Proof (Sketch). By Theorem 6 and a generalisation of Lemma 1 (see [7]), we have that a DQ $\varphi = \bigvee_{j=1}^{m} r_j \geq v_j$ is achievable if and only if there exists π and $\boldsymbol{x} \in \mathbb{R}^m_{\geq 0}$ such that $\forall \sigma \in \Sigma \,.\, \mathbb{E}^{\pi,\sigma}_{\mathcal{G},s}[\boldsymbol{x} \cdot rew(\boldsymbol{r})] \geq \boldsymbol{x} \cdot \boldsymbol{v}$, which is a single-objective query decidable by an NP∩coNP oracle. Given a finite set $X \subseteq \mathbb{R}^m$, we can compute values $d_{\boldsymbol{x}} = \sup_\pi \inf_\sigma \mathbb{E}^{\pi,\sigma}_{\mathcal{G},s}[\boldsymbol{x} \cdot rew(\boldsymbol{r})]$ for all $\boldsymbol{x} \in X$, and define $U_X = \bigcup_{\boldsymbol{x} \in X} \{\boldsymbol{p} \mid \boldsymbol{x} \cdot \boldsymbol{p} \leq d_{\boldsymbol{x}}\}$. It is not difficult to see that U_X yields an under-approximation of achievable points. Let $\tau = \frac{\varepsilon}{2 \cdot m^2 \cdot (M+1)}$. We argue that when we let X be the set of all non-zero vectors $\boldsymbol{x}$ such that $\|\boldsymbol{x}\| = 1$, and where all x_i are of the form $\tau \cdot k_i$ for some $k_i \in \mathbb{N}$, we obtain an ε-approximation of the Pareto set by taking all Pareto points on U_X (see [7] for a proof).

The above approach, together with the algorithm for Pareto set approximations for CQs from Theorem 4, can be used to compute ε-approximations of the Pareto sets for MQs expressed in CNF. The set U_X would then contain tuples of vectors, one per conjunct.

5 Conclusions

We studied stochastic games with multiple expected total reward objectives, and analysed the complexity of the related algorithmic problems. There are several interesting directions for future research. Probably the most obvious is settling the question whether the problem of existence of a strategy achieving a MQ is decidable. Further, it is natural to extend the algorithms to handle long-run objectives containing mean-payoff or ω-regular goals, or to lift the restriction on reward functions to allow both negative and positive rewards at the same time. Another direction is to investigate practical algorithms for the solution for the problems studied here, such as more sophisticated methods for the approximation of Pareto sets.

Acknowledgements. The authors would like to thank Klaus Draeger, Ashutosh Trivedi and Michael Ummels for the discussions about the problem. The authors are partially supported by ERC Advanced Grant VERIWARE, the Institute for the Future of Computing at the Oxford Martin School, EPSRC grant EP/F001096, and the German Academic Exchange Service (DAAD). V. Forejt was suppoted by the Newton Fellowship of Royal Society and is also affiliated with the Faculty of Informatics, Masaryk University, Czech Republic.

References

1. C. Baier, T. Brázdil, M. Größer, and A. Kucera. Stochastic game logic. *Acta Inf.*, 49(4):203–224, 2012.
2. T. Brázdil, V. Brožek, K. Chatterjee, V. Forejt, and A. Kučera. Two views on multiple mean-payoff objectives in Markov decision processes. In *LICS*, 2011.
3. T. Brázdil, V. Brožek, V. Forejt, and A. Kučera. Stochastic games with branching-time winning objectives. In *LICS*, pages 349–358, 2006.
4. K. Chatterjee. *Stochastic Omega-Regular Games*. PhD thesis, EECS Department, University of California, Berkeley, October 2007.
5. K. Chatterjee, R. Majumdar, and T. Henzinger. Markov decision processes with multiple objectives. In *STACS*, pages 325–336. Springer, 2006.
6. K. Chatterjee, M. Randour, and J.-F. Raskin. Strategy synthesis for multi-dimensional quantitative objectives. In *CONCUR*, pages 115–131, 2012.
7. T. Chen, V. Forejt, M. Kwiatkowska, A. Simaitis, and C. Wiltsche. On stochastic games with multiple objectives. Technical Report RR-13-06, Oxford U. DCS, 2013.
8. T. Chen, V. Forejt, M. Z. Kwiatkowska, D. Parker, and A. Simaitis. Automatic verification of competitive stochastic systems. In *TACAS*, pages 315–330, 2012.
9. T. Chen, V. Forejt, M. Z. Kwiatkowska, A. Simaitis, A. Trivedi, and M. Ummels. Playing stochastic games precisely. In *CONCUR*, pages 348–363, 2012.
10. K. Etessami, M. Kwiatkowska, M. Vardi, and M. Yannakakis. Multi-objective model checking of Markov decision processes. *LMCS*, 4(4):1–21, 2008.
11. N. Fijalkow and F. Horn. The surprizing complexity of reachability games. *CoRR*, abs/1010.2420, 2010.
12. V. Forejt, M. Kwiatkowska, G. Norman, D. Parker, and H. Qu. Quantitative multi-objective verification for probabilistic systems. In *TACAS*, 2011.
13. V. Forejt, M. Kwiatkowska, and D. Parker. Pareto curves for probabilistic model checking. In *ATVA*, LNCS, pages 317–332. Springer, 2012.
14. D. Harel. Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness. *J. ACM*, 33(1):224–248, 1986.
15. M. Kattenbelt, M. Z. Kwiatkowska, G. Norman, and D. Parker. A game-based abstraction-refinement framework for markov decision processes. *FMSD*, 2010.
16. M. Kwiatkowska, G. Norman, D. Parker, and H. Qu. Assume-guarantee verification for probabilistic systems. In *TACAS*, pages 23–37. Springer, 2010.
17. R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.
18. D. Martin. The determinacy of Blackwell games. *JSL*, 63(4):1565–1581, 1998.
19. A. Pnueli. Logics and models of concurrent systems. Springer, 1985.
20. L. S. Shapley. Stochastic games. *PNAS*, 39(10):1095, 1953.
21. B. Suman and P. Kumar. A survey of simulated annealing as a tool for single and multiobjective optimization. *J. Oper. Res. Soc.*, 57(10):1143–1160, 2005.
22. Y. Velner, K. Chatterjee, L. Doyen, T. A. Henzinger, A. Rabinovich, and J.-F. Raskin. The complexity of multi-mean-payoff and multi-energy games. *CoRR'12*.