Strategy Synthesis for Stochastic Games with Multiple Long-Run Objectives

Nicolas Basset*, Marta Kwiatkowska*, Ufuk Topcu†, and Clemens Wiltsche*

* Department of Computer Science, University of Oxford, United Kingdom
† Department of Electrical and Systems Engineering, University of Pennsylvania, USA

Abstract. We consider turn-based stochastic games whose winning conditions are conjunctions of satisfaction objectives for long-run average rewards, and address the problem of finding a strategy that almost surely maintains the averages above a given multi-dimensional threshold vector. We show that strategies constructed from Pareto set approximations of expected energy objectives are ε-optimal for the corresponding average rewards. We further apply our methods to compositional strategy synthesis for multi-component stochastic games that leverages composition rules for probabilistic automata, which we extend for long-run ratio rewards with fairness. We implement the techniques and illustrate our methods on a case study of automated compositional synthesis of controllers for aircraft primary electric power distribution networks that ensure a given level of reliability.

1 Introduction

Reactive systems must continually interact with the changing environment. Since it is assumed that they should never terminate, their desirable behaviours are typically specified over infinite executions. Reactive systems are naturally modelled using games, which distinguish between the controllable and uncontrollable events. Stochastic games [13], in particular, allow one to specify uncertainty of outcomes by means of probability distributions. When such models are additionally annotated by rewards that represent, e.g., energy usage and time passage, quantitative objectives and analysis techniques are needed to ensure their correctness. Often, not just a single objective is under consideration, but several, potentially conflicting, objectives must be satisfied, for example maximising both throughput and latency of a network.

In our previous work [6,7], we formulated multi-objective expected total reward properties for stochastic games with certain terminating conditions and showed how ε-optimal strategies can be approximated. Expected total rewards, however, are unable to express long-run average (also called mean-payoff) properties of reactive systems. Another important class of properties are ratio rewards, with which one can state, e.g., speed (distance per time unit) or fuel efficiency (distance per unit of fuel). In this paper we consider controller synthesis for the general class of turn-based stochastic games whose winning conditions are conjunctions of satisfaction objectives for long-run average rewards. We represent

the controllable and uncontrollable actions by Player $\Diamond$ and Player $\Box$, respectively, and address the problem of finding a strategy to satisfy such long-run objectives almost surely for Player $\Diamond$ against all choices of Player $\Box$. These objectives can be used to specify behaviours that guarantee that the probability density is above a threshold, in several dimensions, and the executions actually satisfy the objective we are interested in, which is important for, e.g., reliability and availability analysis. In contrast, expected rewards average the reward over different probabilistic outcomes, possibly with arbitrarily high variance, and thus it may be the case that none of the paths actually satisfy the objective.

Satisfaction Objectives. The specifications we consider are quantitative, in the sense that they are required to maintain the rewards above a certain threshold, and we are interested in almost sure satisfaction, that is, this condition on the rewards is satisfied with probability one. The problem we study generalises the setting of stopping games with multiple satisfaction objectives, which for LTL specifications can be solved via reduction to expected total rewards [7], while our methods are applicable to general turn-based stochastic games. In stopping games, objectives defined using total rewards are appropriate, since existence of the limits is ensured by termination; however, total rewards may diverge for reactive systems, and hence we cannot reduce our problem to total rewards.

Strategy Synthesis. Stochastic games with multiple objectives have been studied in [9], where determinacy under long-run objectives (including ours) is shown (but without strategy construction). However, in general, the winning strategies are history-dependent, requiring infinite memory, which is already the case for Markov decision processes [4]. We restrict to finite memory strategies and utilise the stochastic memory update representation of [6]. For approximating expected total rewards in games, one can construct strategies (in particular, their memory update representation) after finitely many iterations from the difference between achievable values of successive states [7], but long-run properties erase all transient behaviours, and so, in general, we cannot use the achievable values for strategy construction. Inspired by [5], we use expected energy objectives to compute the strategies. These objectives are meaningful in their own right to express that, at every step, the average over some resource requirement does not exceed a certain budget, i.e. some sequences of operations are allowed to violate the budget constraint, as long as they are balanced by other sequences of operations. Consider, for example, sequences of stock market transactions: it is desirable that the expected capital never drops below zero (or some higher value), which can be balanced by credit for individual transactions below the threshold. Synthesis via expected energy objectives yields strategies that not only achieve the required target, but we also obtain a bound on the maximum expected deviation at any step by virtue of the bounded energy. Then, given an achievable target $\boldsymbol{v}$ for mean-payoff, the target $\mathbf{0}$ is ε-achievable by an energy objective with rewards shifted by $-\boldsymbol{v}$, and the same strategy achieves $\boldsymbol{v} - \varepsilon$ for the mean-payoff objective under discussion.

Compositional Synthesis. In our previous work [3], we proposed a synchronising parallel composition for stochastic games that enables a compositional

approach to controller synthesis that significantly outperforms the monolithic method. The strategy for the composition of games is derived from the strategies synthesised for the individual components. To apply these methods for a class of objectives (e.g. total rewards), one must (i) show that the objectives are defined on traces, i.e. synchronisation of actions is sufficient for information sharing; (ii) provide compositional verification rules for probabilistic automata (e.g. assume-guarantee rules); and (iii) provide synthesis methods for single component games. We address these points for long-run average objectives, extending [10] for (ii), enabling compositional synthesis for ratio rewards. A key characteristic of the rules is the use of fairness, which requires that no component is prevented from making progress. The methods of [3] were presented with total rewards, where (trivial) fairness was only guaranteed through synchronised termination.

Case Study. We implement the methods and demonstrate their scalability and usefulness via a case study that concerns the control of the electric power distribution on aircraft [11]. In avionics, the transition to more-electric aircraft has been brought about by advances in electronics technology, reducing take-off weight and power consumption. We extend the (non-quantitative) game-theoretic approach of [16] to the stochastic games setting with multiple long-run satisfaction objectives, where the behaviour of generators is described stochastically. We demonstrate how our approach yields controllers that ensure given reliability levels and higher uptimes than those reported in [16].

Contributions. Our main contributions are as follows.

- We show that expected energy objectives enable synthesis of ε-optimal finite-memory strategies for almost sure satisfaction of average rewards (Theorem 2).
- We propose a semi-algorithm to construct ε-optimal strategies using stochastically updated memory (Theorem 1).
- We extend compositional rules to specifications defined on traces, and hence show how to utilise ratio rewards in compositional synthesis (Theorem 3).
- We demonstrate compositional synthesis using long-run objectives via a case study of an aircraft electric power distribution network.

Related Work. For Markov decision processes (MDPs), multi-dimensional long-run objectives for satisfaction and expectation were studied in [4], and expected ratio rewards in [15]. Satisfaction for long-run properties in stochastic games is the subject of [9]; in particular, they present algorithms for combining a single mean-payoff with a Büchi objective, which rely on the non-quantitative nature of the Büchi objective, and hence cannot be straightforwardly extended to several mean-payoff objectives that we consider. Non-stochastic games with energy objectives have been considered, for example, in [5], where it is assumed that Player $\Box$ plays deterministically, in contrast to our approach that permits the use of stochasticity. Our almost sure satisfaction objectives are related to the concept of quantiles in [1], in that they correspond to 1-quantiles, but here we consider mean-payoff objectives for games. An extended version of this paper, including proofs, can be found in [2].

2 Preliminaries

Notation. A *discrete probability distribution* (or *distribution*) over a (countable) set Q is a function $\mu : Q \to [0,1]$ such that $\sum_{q \in Q} \mu(q) = 1$; its *support* $\mathsf{supp}(\mu)$ is $\{q \in Q \mid \mu(q) > 0\}$. We denote by $\mathcal{D}(Q)$ the set of all distributions over Q with finite support. A distribution $\mu \in \mathcal{D}(Q)$ is *Dirac* if $\mu(q) = 1$ for some $q \in Q$, and if the context is clear we just write q to denote such a distribution μ.

We work with the usual metric-space topology on $\mathbb{R}^n$. The *downward closure* of a set X is defined as $\mathsf{dwc}(X) \stackrel{\text{def}}{=} \{\boldsymbol{y} \mid \exists \boldsymbol{x} \in X \,.\, \boldsymbol{y} \leq \boldsymbol{x}\}$. A set $X \subseteq \mathbb{R}^n$ is *convex* if for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in X$, and all $\alpha \in [0,1]$, $\alpha\boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2 \in X$; its *convex hull* $\mathsf{conv}(X)$ is the smallest convex set containing X. Given a set X, $\alpha \times X$ denotes the set $\{\alpha \cdot \boldsymbol{x} \mid \boldsymbol{x} \in X\}$. The *Minkowski sum* of sets X and Y is $X + Y \stackrel{\text{def}}{=} \{\boldsymbol{x} + \boldsymbol{y} \mid \boldsymbol{x} \in X, \boldsymbol{x} \in Y\}$. We refer to the sth component of a vector $\boldsymbol{v}$ by v_s and $[\boldsymbol{v}]_s$. We write $\boldsymbol{\varepsilon}$ to denote the vector $(\varepsilon, \varepsilon, \ldots, \varepsilon)$. For a vector $\boldsymbol{x}$ (resp. vector of sets Z) and a scalar ε, define $\boldsymbol{x} + \varepsilon$ by $[\boldsymbol{x} + \varepsilon]_s = x_s + \varepsilon$ (resp. $[Z + \varepsilon]_s \stackrel{\text{def}}{=} Z_s + \varepsilon$) for all components s of $\boldsymbol{x}$ (resp. Z), where, for a set X, let $X + \varepsilon \stackrel{\text{def}}{=} \{\boldsymbol{x} + \varepsilon \mid \boldsymbol{x} \in X\}$. For vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, $\boldsymbol{x} \cdot \boldsymbol{y}$ denotes their dot-product, and $\boldsymbol{x} \bullet \boldsymbol{y}$ denotes component-wise multiplication.

Stochastic Games. We consider turn-based action-labelled stochastic two-player games (henceforth simply called *games*), which distinguish two types of nondeterminism, each controlled by a separate player. Player $\Diamond$ represents the controllable part for which we want to synthesise a strategy, while Player $\Box$ represents the uncontrollable environment.

Definition 1. *A* game G *is a tuple* $\langle S, (S_\Diamond, S_\Box), \varsigma_0, \mathcal{A}, \longrightarrow \rangle$*, where* S *is a finite set of* states *partitioned into* Player $\Diamond$ states $S_\Diamond$ *and* Player $\Box$ states $S_\Box$*;* $\varsigma_0 \in S$ *is an* initial state*;* $\mathcal{A}$ *is a finite set of* actions*; and* $\longrightarrow \subseteq S \times (\mathcal{A} \cup \{\tau\}) \times \mathcal{D}(S)$ *is a* transition relation*, such that, for all* s*,* $\{(s, a, \mu) \in \longrightarrow\}$ *is finite.*

We write $s \xrightarrow{a} \mu$ for a *transition* $(s, a, \mu) \in \longrightarrow$. The action labels $\mathcal{A}$ on transitions model observable behaviours, whereas τ can be seen as internal: it cannot be used in winning conditions and is not synchronised in the composition. We denote the set of *moves* (also called *stochastic states*) by $S_\bigcirc \stackrel{\text{def}}{=} \{(a, \mu) \in \mathcal{A} \times \mathcal{D}(S) \mid \exists s \in S \,.\, s \xrightarrow{a} \mu\}$, and let $\overline{S} = S \cup S_\bigcirc$. Let the set of *successors* of $s \in \overline{S}$ be $\mathsf{succ}(s) \stackrel{\text{def}}{=} \{(a, \mu) \in S_\bigcirc \mid s \xrightarrow{a} \mu\} \cup \{t \in S \mid \mu(t) > 0 \text{ with } s = (a, \mu)\}$. A *probabilistic automaton* (PA, [12]) is a game with $S_\Diamond = \emptyset$, and a *discrete-time Markov chain* (DTMC) is a PA with $|\mathsf{succ}(s)| = 1$ for all $s \in S$.

A finite (infinite) *path* $\lambda = s_0(a_0, \mu_0)s_1(a_1, \mu_1)s_2 \ldots$ is a finite (infinite) sequence of alternating states and moves, such that for all $i \geq 0$, $s_i \xrightarrow{a_i} \mu_i$ and $\mu_i(s_{i+1}) > 0$. A finite path λ ends in a state, denoted $\mathsf{last}(\lambda)$. A finite (infinite) *trace* is a finite (infinite) sequence of actions. Given a path, its trace is the sequence of actions along λ, with τ projected out. Formally, $\mathsf{trace}(\lambda) \stackrel{\text{def}}{=} \textsc{proj}_{\{\tau\}}(a_0 a_1 \ldots)$, where, for $\alpha \subseteq \mathcal{A} \cup \{\tau\}$, $\textsc{proj}_\alpha$ is the morphism defined by $\textsc{proj}_\alpha(a) = a$ if $a \notin \alpha$, and ϵ (the empty trace) otherwise.

Strategies. Nondeterminism for each player is resolved by a strategy, which maps finite paths to distributions over moves. For PAs, we do not speak of player

strategies, and implicitly consider strategies of Player $\Box$. Here we use an alternative, equivalent formulation of strategies using stochastic memory update [4].

Definition 2. *A Player $\Diamond$ strategy π is a tuple $\langle \mathfrak{M}, \pi_u, \pi_c, \alpha \rangle$, where $\mathfrak{M}$ is a countable set of* memory elements*; $\pi_u \colon \mathfrak{M} \times S \to \mathcal{D}(\mathfrak{M})$ is a* memory update function*; $\pi_c \colon S_\Diamond \times \mathfrak{M} \to \mathcal{D}(S)$ is a* next move function *s.t. $\pi_c(s,m)(t) > 0$ only if $t \in \mathsf{succ}(s)$; and $\alpha \colon S \to \mathcal{D}(\mathfrak{M})$ defines for each state of G an* initial memory distribution*. A Player $\Box$ strategy σ is defined in an analogous manner.*

A strategy is *finite-memory* if $|\mathfrak{M}|$ is finite. Applying a strategy pair (π, σ) to a game G yields an *induced DTMC* $G^{\pi,\sigma}$ [7]; an induced DTMC contains only reachable states and moves, but retains the entire action alphabet of G.

Probability Measures and Expectations. The *cylinder set* of a finite path λ (resp. finite trace $w \in \mathcal{A}^*$) is the set of infinite paths (resp. traces) with prefix λ (resp. w). For a finite path $\lambda = s_0(a_0,\mu_0)s_1(a_1,\mu_1)\dots s_n$ in a DTMC D we define $\Pr_{D,s_0}(\lambda)$, the measure of its cylinder set, by $\Pr_{D,s_0}(\lambda) \stackrel{\text{def}}{=} \prod_{i=0}^{n-1} \mu_i(s_{i+1})$, and write $\Pr_{G,s}^{\pi,\sigma}$ for $\Pr_{G^{\pi,\sigma},s}$. For a finite trace w, $\mathsf{paths}(w)$ denotes the set of minimal finite paths with trace w, i.e. $\lambda \in \mathsf{paths}(w)$ if $\mathsf{trace}(\lambda) = w$ and there is no path $\lambda' \neq \lambda$ with $\mathsf{trace}(\lambda') = w$ and λ' being a prefix of λ. The measure of the cylinder set of w is $\tilde{\Pr}_{D,s}(w) \stackrel{\text{def}}{=} \sum_{\lambda \in \mathsf{paths}(w)} \Pr_{D,s}(\lambda)$, and we call $\tilde{\Pr}_{D,s}$ the *trace distribution* of D. The measures uniquely extend to infinite paths due to Carathéodory's extension theorem. We denote the set of infinite paths of D starting at s by $\Omega_{D,s}$. The *expectation* of a function $\boldsymbol{\rho} : \Omega_{D,s} \to \mathbb{R}^n_{\pm\infty}$ over infinite paths in a DTMC D is $\mathbb{E}_{D,s}[\boldsymbol{\rho}] \stackrel{\text{def}}{=} \int_{\lambda \in \Omega_{D,s}} \boldsymbol{\rho}(\lambda) d\Pr_{D,s}(\lambda)$.

Rewards. A *reward structure* (with n-dimensions) of a game is a partial function $r : \overline{S} \to \mathbb{R}$ ($\boldsymbol{r} : \overline{S} \to \mathbb{R}^n$). A reward structure r is *defined on actions* $\mathcal{A}_r$ if $r(a,\mu) = r(a,\mu')$ for all moves $(a,\mu),(a,\mu') \in S_{\bigcirc}$ such that $a \in \mathcal{A}_r$, and $r(s) = 0$ otherwise; and if the context is clear we consider it as a total function $r : \mathcal{A}_r \to \mathbb{R}$ for $\mathcal{A}_r \subseteq \mathcal{A}$. Given an n-dimensional reward structure $\boldsymbol{r} : \overline{S} \mapsto \mathbb{R}^n$, and a vector $\boldsymbol{v} \in \mathbb{R}^n$, define the reward structure $\boldsymbol{r} - \boldsymbol{v}$ by $[\boldsymbol{r} - \boldsymbol{v}]_s \stackrel{\text{def}}{=} \boldsymbol{r}(s) - \boldsymbol{v}$ for all $s \in \overline{S}$. For a path $\lambda = s_0 s_1 \dots$ and a reward structure r we define $\mathrm{rew}^N(r)(\lambda) \stackrel{\text{def}}{=} \sum_{i=0}^{N} r(s_i)$, for $N \geq 0$; the *average reward* is $\mathrm{mp}(r)(\lambda) \stackrel{\text{def}}{=} \liminf_{N\to\infty} \frac{1}{N+1}\mathrm{rew}^N(r)(\lambda)$; given a reward structure c such that, for all $s \in \overline{S}$, $c(s) \geq 0$ and, for all bottom strongly connected components (BSCCs) $\mathcal{B}$ of D, there is a state s in $\mathcal{B}$ such that $c(s) > 0$, the *ratio reward* is $\mathrm{ratio}(r/c)(w) \stackrel{\text{def}}{=} \liminf_{N\to\infty} \mathrm{rew}^N(r)(w)/(1+\mathrm{rew}^N(c)(w))$. If D has finite state space, the $\liminf$ of the above rewards can be replaced by the true limit in the expectation, as it is almost surely defined. Further, the above rewards straightforwardly extend to multiple dimensions using vectors.

Specifications and Objectives. A *specification* φ is a predicate on path distributions, and we write $D \models \varphi$ if $\varphi(\Pr_{D,\varsigma_0})$ holds. We say that a Player $\Diamond$ strategy π *wins* for a specification φ in a game G, written $\pi \models \varphi$, if, for all Player $\Box$ strategies σ, $G^{\pi,\sigma} \models \varphi$, and say that φ is *achievable* if such a winning strategy exists. A specification φ is *defined on traces of* $\mathcal{A}$ if $\varphi(\tilde{\Pr}_{D,\varsigma_0}) = \varphi(\tilde{\Pr}_{D',\varsigma'_0})$ for all DTMCs D, D' such that $\tilde{\Pr}_{D,\varsigma_0}(w) = \tilde{\Pr}_{D',\varsigma'_0}(w)$ for all traces $w \in \mathcal{A}^*$.

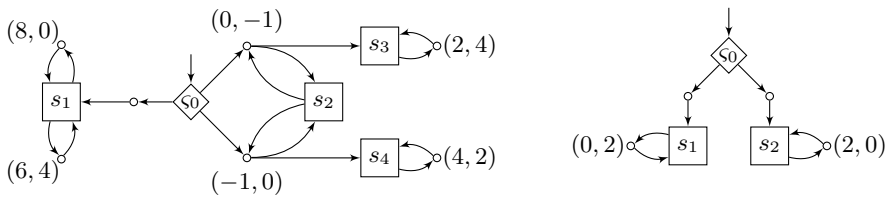


Fig. 1: Example games. Moves and states for Player ◊ and Player □ are shown as ○, ◊ and □ resp.; two-dimensional rewards shown where non-zero.

A DTMC D satisfies an *expected energy* specification $\mathrm{EE}_s(\boldsymbol{r})$ if there exists $\boldsymbol{v}_0$ such that $\mathbb{E}_{D,s}[\mathrm{rew}^N(\boldsymbol{r})] \geq \boldsymbol{v}_0$ for all $N \geq 0$; D satisfies $\mathrm{EE}(\boldsymbol{r})$ if, for every state s of D, D satisfies $\mathrm{EE}_s(\boldsymbol{r})$. An *almost sure average* (resp. *ratio*) *reward objective* for target v is $\mathrm{Pmp}_s(\boldsymbol{r})(\boldsymbol{v}) \equiv \Pr_{D,s}(\mathrm{mp}(\boldsymbol{r}) \geq \boldsymbol{v}) = 1$ (resp. $\mathrm{Pratio}_s(\boldsymbol{r})(\boldsymbol{v}) \equiv \Pr_{D,s}(\mathrm{ratio}(\boldsymbol{r}/\boldsymbol{c}) \geq \boldsymbol{v}) = 1$). If the rewards $\boldsymbol{r}$ and $\boldsymbol{c}$ are understood, we omit them and write just $\mathrm{Pmp}_s(\boldsymbol{v})$ and $\mathrm{Pratio}_s(\boldsymbol{v})$. By using n-dimensional reward structures, we require that a strategy achieves the *conjunction* of the objectives defined on the individual dimensions. Minimisation is supported by inverting signs of rewards. Given an objective φ with target vector $\boldsymbol{v}$, denote by $\varphi[\boldsymbol{x}]$ the objective φ with $\boldsymbol{v}$ substituted by $\boldsymbol{x}$. A target $\boldsymbol{v} \in \mathbb{R}^n$ is a *Pareto vector* if $\varphi[\boldsymbol{v} - \varepsilon]$ is achievable for all $\varepsilon > 0$, and $\varphi[\boldsymbol{v} + \varepsilon]$ is not achievable for any $\varepsilon > 0$. The downward closure of the set of all such vectors is called a *Pareto set*.

Example. Consider the game in Figure 1 (left), showing a stochastic game with a two-dimensional reward structure. Player ◊ can achieve $\mathrm{Pmp}_{\varsigma_0}(3, 0)$ if going left at ς_0, and $\mathrm{Pmp}_{\varsigma_0}(1, 1)$ if choosing either move to the right, since then s_3 and s_4 are almost surely reached. Furthermore, achieving an expected mean-payoff does not guarantee achieving almost-sure satisfaction in general: the Player ◊ strategy going up right from ς_0 achieves an expected mean-payoff of at least $(1, 1.5)$, which by the above argument cannot be achieved almost surely. Also, synthesis in MDPs [4,15] can utilise the fact that the strategy controls reachability of end-components; e.g., if all states in the game of Figure 1 (left) are controlled by Player ◊, $(3, 2)$ is almost surely achievable.

3 Strategy Synthesis for Average Rewards

We consider the problem of computing ε-optimal strategies for almost sure average reward objectives $\mathrm{Pmp}_{\varsigma_0}(\boldsymbol{v})$. Note that, for any $\boldsymbol{v} \geq 0$, the objective $\mathrm{Pmp}_{\varsigma_0}(\boldsymbol{r})(\boldsymbol{v})$ is equivalent to $\mathrm{Pmp}_{\varsigma_0}(\boldsymbol{r} - \boldsymbol{v})(\boldsymbol{0})$, i.e. with the rewards shifted by $-\boldsymbol{v}$. Hence, from now on we assume w.l.o.g. that the objectives have target $\boldsymbol{0}$.

3.1 Expected Energy Objectives

We show how synthesis for almost sure average reward objectives reduces to synthesis for expected energy objectives. Applying finite-memory strategies to

games results in finite induced DTMCs. Infinite memory may be required for winning strategies of Player $\Diamond$ [4]; here we synthesise only finite-memory strategies for Player $\Diamond$, in which case only finite memory for Player $\Box$ is sufficient:

Lemma 1. *A finite-memory Player $\Diamond$ strategy is winning for the objective* $\mathrm{EE}(\boldsymbol{r})$ *(resp.* $\mathrm{Pmp}_{\varsigma_0}(\boldsymbol{r})(\boldsymbol{v})$*) if it wins against all finite-memory Player $\Box$ strategies.*

We now state our key reduction lemma to show that almost sure average reward objectives can be ε-approximated by considering EE objectives.

Lemma 2. *Given a finite-memory strategy π for Player $\Diamond$, the following hold:*

(i) if π satisfies $\mathrm{EE}(\boldsymbol{r})$*, then π satisfies* $\mathrm{Pmp}_{\varsigma_0}(\boldsymbol{r})(\mathbf{0})$*; and*
(ii) if π satisfies $\mathrm{Pmp}_{\varsigma_0}(\boldsymbol{r})(\mathbf{0})$*, then, for all $\varepsilon > 0$, π satisfies* $\mathrm{EE}(\boldsymbol{r} + \boldsymbol{\varepsilon})$*.*

Our method described in Theorem 2 below allows us to compute $\mathrm{EE}(\boldsymbol{r} + \boldsymbol{\varepsilon})$, and hence, by virtue of Lemma 2(i), derive ε-optimal strategies for $\mathrm{Pmp}_{\varsigma_0}(\mathbf{0})$. Item (ii) of Lemma 2 guarantees completeness of our method, in the sense that, for any vector $\boldsymbol{v}$ such that $\mathrm{Pmp}_{\varsigma_0}(\boldsymbol{r})(\boldsymbol{v})$ is achievable, we compute an ε-optimal strategy; however, if $\boldsymbol{v}$ is not achievable, our algorithm does not terminate.

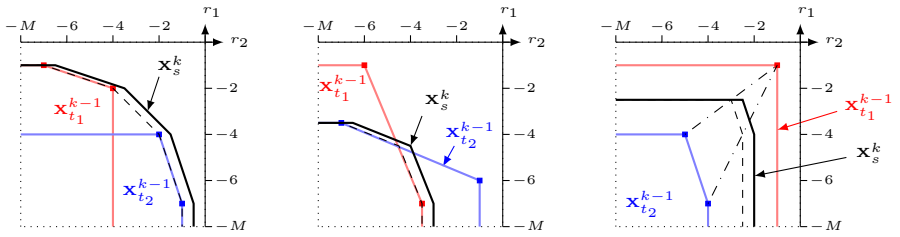
3.2 Strategy Construction

We define a value iteration method that in k iterations computes the sets X_s^k of shortfall vectors at state s, so that for any $\boldsymbol{v}_0 \in X_s^k$, Player $\Diamond$ can keep the expected energy above $\boldsymbol{v}_0$ during k steps of the game. Moreover, if successive sets X_s^{k+1} and X_s^k satisfy $X_s^k \sqsubseteq X_s^{k+1} + \varepsilon$, where $A \sqsubseteq B \Leftrightarrow \mathsf{dwc}(A) \subseteq \mathsf{dwc}(B)$, then we can construct a finite-memory strategy for $\mathrm{EE}(\boldsymbol{r} + \varepsilon)$ using Theorem 1.

Value Iteration. Let $\mathrm{Box}_M \stackrel{\text{def}}{=} [-M, 0]^n$. The *$M$-downward closure* of a set X is $\mathrm{Box}_M \cap \mathsf{dwc}(X)$. Let $\mathcal{P}_c^M(X)$ be the set of convex closed M-downward-closed subsets of X. Let $\mathcal{L}_M \stackrel{\text{def}}{=} (\mathcal{P}_c^M(\mathrm{Box}_M))^{|\overline{S}|}$, endow it with the partial order $X \subseteq Y \Leftrightarrow \forall s \in \overline{S} \,.\, X_s \subseteq Y_s$, and add the *top element* $\top \stackrel{\text{def}}{=} \mathrm{Box}_M^{|\overline{S}|}$. For a fixed M, define the operator $F_M : \mathcal{L}_M \to \mathcal{L}_M$ by $[F_M(X)]_s \stackrel{\text{def}}{=} \mathrm{Box}_M \cap \mathsf{dwc}(Y_s)$, where

$$Y_s \stackrel{\text{def}}{=} \boldsymbol{r}(s) + \begin{cases} \mathsf{conv}(\bigcup_{t \in \mathsf{succ}(s)} X_t) & \text{if } s \in S_\Diamond \\ \bigcap_{t \in \mathsf{succ}(s)} X_t & \text{if } s \in S_\Box \\ \sum_{t \in \mathsf{supp}(\mu)} \mu(t) \times X_t & \text{if } s = (a, \mu) \in S_\bigcirc. \end{cases}$$

The operator F_M reflects what Player $\Diamond$ can achieve in the respective state types. In $s \in S_\Diamond$, Player $\Diamond$ can achieve the values in successors (union), and can randomise between them (convex hull). In $s \in S_\Box$, Player $\Diamond$ can achieve only values that are in all successors (intersection), since Player $\Box$ can pick arbitrarily. Lastly, in $s \in S_\bigcirc$, Player $\Diamond$ can achieve values with the prescribed distribution. F_M is closely related to our operator for expected total rewards in [6], but here we cut off values above zero with Box_M, similarly to the controllable predecessor operator of [5] for computing energy in non-stochastic games. Box_M ensures that the



(a) Player ◊ state s. (b) Player □ state s. (c) Move $s \in S_{\bigcirc}$.

Fig. 2: Value iteration and strategy construction, for state s with successors t_1, t_2, and reward $r_1(s) = 0.5$, $r_2(s) = 0$. The Pareto set under-approximation X_s^k is computed from $X_{t_1}^{k-1}$ and $X_{t_2}^{k-1}$. To achieve a point $\boldsymbol{p} \in C_s^k$, the strategy updates its memory as follows: for $s \in S_{\Box}$, for all $t \in \mathsf{succ}(s)$, $\boldsymbol{p} - \boldsymbol{r}(s) \in \mathsf{conv}(C_t^{k-1})$; for $s \in S_{\Diamond} \cup S_{\bigcirc}$, there exist successors $t \in \mathsf{succ}(s)$ and a distribution α s.t. $\boldsymbol{p} - \boldsymbol{r}(s) \in \sum_t \alpha(t) \times \mathsf{conv}(C_t^k)$, where, for $s = (a, \mu) \in S_{\bigcirc}$, we fix $\alpha = \mu$. As F is order preserving, it is sufficient to use X_t^l instead of X_t^k for any $l \geq k$.

strategy we construct in Theorem 1 below never allows the energy to diverge in any reachable state. For example, in Figure 1 (right), for $\boldsymbol{v} = (\frac{1}{2}, \frac{1}{2})$, $\mathrm{EE}_{\varsigma_0}(\boldsymbol{r} - \boldsymbol{v})$ is achievable while, for the states $s \in \{s_1, s_2\}$, $\mathrm{EE}_s(\boldsymbol{r} - \boldsymbol{v})$ is not. Since one of s_1 or s_2 must be reached, $\mathrm{EE}(\boldsymbol{r} - \boldsymbol{v})$ is not achievable, disallowing the use of Lemma 2(i); and indeed, $\mathrm{Pmp}_{\varsigma_0}(\boldsymbol{v})$ is not achievable. Bounding with M allows us to use a geometric argument in Lemma 3 below, replacing the finite lattice arguments of [5], since our theory is more involved as it reflects the continuous essence of randomisation.

We show in the following proposition that F_M defines a monotonic fixpoint computation and that it converges to the greatest fixpoint of F_M. Its proof relies on Scott-continuity of F_M, and invokes the Kleene fixpoint theorem.

Proposition 1. *F_M is order-preserving, $\top \supseteq F_M(\top) \supseteq F_M^2(\top) \supseteq \cdots$, and the greatest fixpoint $fix(F_M)$ exists and is equal to $\lim_{k\to\infty} F_M^k(\top) = \cap_{k\geq 0} F_M^k(\top)$.*

Further, we use F_M to compute the set of shortfall vectors required for Player ◊ to win for $\mathrm{EE}_s(\boldsymbol{r})$ via a value iteration with relative stopping criterion defined using ε, see Lemma 3 below. Denote $X^k \stackrel{\text{def}}{=} F_M^k(\top)$. The value iteration is illustrated in Figure 2: at iteration k, the set X_s^k of possible shortfalls until k steps is computed from the corresponding sets X_t^{k-1} for successors $t \in \mathsf{succ}(s)$ of s at iteration $k-1$. The values are restricted to be within Box_M, so that obtaining an empty set at a state s in the value iteration is an indicator of divergence at s. Any state that must be avoided by Player ◊ yields an empty set. For instance, in Figure 1 (left), with target $(1, 1)$ the value iteration diverges at s_1 for any $M \geq 0$, but at ς_0, Player ◊ can go to the right to avoid accessing s_1. The following proposition ensures completeness of our method, stated in Theorem 2 below.

Proposition 2. *If $\mathrm{EE}(\boldsymbol{r})$ is achievable then $[fix(F_M)]_{\varsigma_0} \neq \emptyset$ for some $M \geq 0$.*

Proof (Sketch). First, we consider the expected enrgy of finite DTMCs, where, at every step, we cut off the positive values. This entails that the sequence of the resulting truncated non-positive expected energies decreases and converges toward a limit vector $\boldsymbol{u}$ whose coordinates are finite if $\mathrm{EE}(\boldsymbol{r})$ is satisfied. We show that, when $\mathrm{EE}(\boldsymbol{r})$ is satisfied by a strategy π, there is a global lower bound $-M$ on every coordinate of the limit vector $\boldsymbol{u}$ for the DTMC $G^{\pi,\sigma}$ induced by any $\mathsf{Player}\ \Box$ strategy σ. We show that, for this choice of M, the fixpoint of F_M for the game G is non-empty in every state reachable under π. We conclude that $[fix(F_M)]_{\varsigma_0} \neq \emptyset$ for some $M \geq 0$ whenever $\mathrm{EE}(\boldsymbol{r})$ is achievable.

Lemma 3. *Given M and ε, for every non-increasing sequence (X^i) of elements of $\mathcal{L}_M$ there exists $k \leq k^{**} \stackrel{\text{def}}{=} \left[2n((\lceil \frac{M}{\varepsilon} \rceil + 2)^2 + 2)\right]^{|\overline{S}|}$ such that $X^k \sqsubseteq X^{k+1} + \varepsilon$.*

Proof (Sketch). We first consider a single state s, and construct a graph with vertices from the sequence of sets (X^i), and edges indicating dimensions where the distance is at least ε. Interpreting each dimension as a colour, we use a Ramseyan argument to find the bound $k^* \stackrel{\text{def}}{=} n \cdot ((\lceil \frac{M}{\varepsilon} \rceil + 2)^2 + 2)$ for a single state. To find the bound $k^{**} \stackrel{\text{def}}{=} (2k^*)^{|\overline{S}|}$, which is for *all* states, we extract successive subsequences of $\{1, 2, \ldots, k^{**}\} \stackrel{\text{def}}{=} I_0 \supseteq I_1 \supseteq \cdots \supseteq I_{|\overline{S}|}$, where going from I_i to I_{i+1} means that one additional state has the desired property, and such that the invariant $|I_{i+1}| \geq |I_i|/(2k^*)$ is satisfied. At the end $I_{|\overline{S}|}$ contains at least one index $k \leq k^{**}$ for which all states have the desired property.

Strategy Construction. The strategies are constructed so that their memory corresponds to the extreme points of the sets computed by $F_M^k(\top)$. The strategies stochastically update their memory, and so the expectation of their memory elements corresponds to an expectation over such extreme points.

Let C_s^k be the set of *extreme points* of $\mathsf{dwc}(X_s^k)$, for all $k \geq 0$ (since $X^k \in \mathcal{L}_M$, the sets X_s^k are closed). For any point $\boldsymbol{p} \in X_s^k$, there is some $\boldsymbol{q} \geq \boldsymbol{p}$ that can be obtained by a convex combination of points in C_s^k, and so the strategy we construct uses C_s^k as memory, randomising to attain the convex combination $\boldsymbol{q}$. Note that the sets C_s^k are finite, yielding finite-memory strategies.

If $X_{\varsigma_0}^{k+1} \neq \emptyset$ and $X^k \sqsubseteq X^{k+1} + \varepsilon$ for some $k \in \mathbb{N}$ and $\varepsilon \geq 0$, we can construct a $\mathsf{Player}\ \Diamond$ strategy π for $\mathrm{EE}(\boldsymbol{r} + \varepsilon)$. Denote by $\overline{T} \subseteq \overline{S}$ the set of states s for which $X_s^{k+1} \neq \emptyset$. For $l \geq 1$, define the *standard l-simplex* by $\Delta^l \stackrel{\text{def}}{=} \{B \in [0,1]^l \mid \sum_{\beta \in B} \beta = 1\}$. The memory $\mathfrak{M} \stackrel{\text{def}}{=} \bigcup_{s \in \overline{T}} \{(s, \boldsymbol{p}) \mid \boldsymbol{p} \in C_s^k\}$ is initialised according to α, defined by $\alpha(s) \stackrel{\text{def}}{=} [(s, \boldsymbol{q}_0^s) \mapsto \beta_0^s, \ldots, (s, \boldsymbol{q}_n^s) \mapsto \beta_n^s]$, where $\boldsymbol{\beta}^s \in \Delta^n$, and, for all $1 \leq i \leq n$, $\boldsymbol{q}_i^s \in C_s^k$. The update π_u and next move function π_c are defined as follows: at state s with memory $(s, \boldsymbol{p})$, for all $t \in \mathsf{succ}(s)$, pick n vectors $\boldsymbol{q}_i^t \in C_t^k$ for $1 \leq i \leq n$, with coefficients $\boldsymbol{\beta}^t \in \Delta^n$, such that

- **for** $s \in S_\Diamond$**,** there is $\boldsymbol{\gamma} \in \Delta^{|\mathsf{succ}(s) \cap \overline{T}|}$, such that $\sum_t \gamma_t \cdot \sum_i \beta_i^t \cdot \boldsymbol{q}_i^t \geq \boldsymbol{p} - \boldsymbol{r}(s) - \varepsilon$;
- **for** $s \in S_\Box$**,** for all $t \in \mathsf{succ}(s)$, $\sum_i \beta_i^t \cdot \boldsymbol{q}_i^t \geq \boldsymbol{p} - \boldsymbol{r}(s) - \varepsilon$; and
- **for** $s = (a, \mu) \in S_\bigcirc$**,** we have $\sum_{t \in \mathsf{supp}(\mu)} \mu(t) \cdot \sum_i \beta_i^t \cdot \boldsymbol{q}_i^t \geq \boldsymbol{p} - \boldsymbol{r}(s) - \varepsilon$;

Algorithm 1 PMP Strategy Synthesis

```
1: function SynthPMP(G, r, v, ε)
2:     Set the reward structure to r − v + ε/2; let k ← 0; M ← 2; X^0 ← ⊤;
3:     while true do
4:         while X^k ⋢ X^{k+1} + ε/2 do
5:             k ← k + 1; X^{k+1} ← F_M(X^k);
6:         if X^k_{ς0} ≠ ∅ then
7:             Construct π for ε/2 and any v_0 ∈ C^k_{ς0} using Theorem 1; return π
8:         else
9:             k ← 0; M ← M^2;
```

and, for all $t \in \mathsf{succ}(s)$, let $\pi_u((s,\boldsymbol{p}),t)(t,\boldsymbol{q}_i^t) \stackrel{\text{def}}{=} \beta_i^t$ for all i, and $\pi_c(s,(s,\boldsymbol{p}))(t) \stackrel{\text{def}}{=} \gamma_t$ if $s \in S_\Diamond$.

Theorem 1. *If $X^{k+1}_{\varsigma_0} \neq \emptyset$ and $X^k \sqsubseteq X^{k+1} + \varepsilon$ for some $k \in \mathbb{N}$ and $\varepsilon \geq 0$, then the Player $\Diamond$ strategy constructed above is finite-memory and wins for* $\mathrm{EE}(\boldsymbol{r}+\boldsymbol{\varepsilon})$.

Proof (Sketch). We show the strategy is well-defined, i.e. the relevant extreme points and coefficients exist, which is a consequence of $X^k \sqsubseteq X^{k+1} + \varepsilon$. We then show that, when entering a state s_o with a memory $\boldsymbol{p}_o$, the expected memory from this state after N steps is above $\boldsymbol{p}_o - \mathbb{E}_{D,s_o}[\mathrm{rew}^N(\boldsymbol{r})] - N\boldsymbol{\varepsilon}$. As the memory is always non-positive, this implies that $\mathbb{E}_{D,s_o}[\mathrm{rew}^N(\boldsymbol{r}+\boldsymbol{\varepsilon})] \geq \boldsymbol{p}_o \geq -M$ for every state s_o with memory $\boldsymbol{p}_o$, for every N. We conclude that $\mathrm{EE}(\boldsymbol{r}+\boldsymbol{\varepsilon})$ holds.

3.3 Strategy Synthesis Algorithm

Given a game G, a reward structure $\boldsymbol{r}$ with target vector $\boldsymbol{v}$, and $\varepsilon > 0$, the semi-algorithm given in Algorithm 1 computes a strategy winning for $\mathrm{Pmp}_{\varsigma_0}(\boldsymbol{r})(\boldsymbol{v}-\varepsilon)$.

Theorem 2. *Whenever $\boldsymbol{v}$ is in the Pareto set of* $\mathrm{Pmp}_{\varsigma_0}(\boldsymbol{r})$*, then Algorithm 1 terminates with a finite-memory ε-optimal strategy.*

Proof (Sketch). Since $\boldsymbol{v}$ is in the Pareto set of the almost sure average reward objective, by Lemma 2(ii) the objective $\mathrm{EE}(\boldsymbol{r}-\boldsymbol{v}+\frac{\boldsymbol{\varepsilon}}{2})$ is achievable, and, by Proposition 2, there exists an M such that $fix(F_M)$ is nonempty. The condition in Line 6 is then satisfied as $\emptyset \neq [fix(F_M)]_{\varsigma_0} \subseteq X^k_{\varsigma_0}$. Further, due to the bound M on the size of the box Box_M in the value iteration, the inner loop terminates after a finite number of steps, as shown in Lemma 3. Then, by Theorem 1, the strategy constructed in Line 7 (with degradation factor $\frac{\varepsilon}{2}$ for the reward $\boldsymbol{r}-\boldsymbol{v}+\frac{\boldsymbol{\varepsilon}}{2}$) satisfies $\mathrm{EE}(\boldsymbol{r}-\boldsymbol{v}+\boldsymbol{\varepsilon})$, and hence, using Lemma 2(i), $\mathrm{Pmp}_{\varsigma_0}(\boldsymbol{r})(\boldsymbol{v}-\boldsymbol{\varepsilon})$.

4 Compositional Synthesis

In order to synthesise strategies compositionally, we introduced in [3] a composition of games, and showed that assume-guarantee rules for PAs can be applied in

synthesis for games: whenever there is a PA verification rule, the corresponding game synthesis rule has the same form and side-conditions (Theorem 1 of [3]). We present a PA assume-guarantee rule for ratio rewards. The PA rules in [10] only support total expected rewards, while our rule works with any specification defined on traces, and in particular with ratio rewards (Proposition 4).

Ratio Rewards. Ratio rewards $\mathrm{ratio}(\boldsymbol{r}/\boldsymbol{c})$ generalise average rewards $\mathrm{mp}(\boldsymbol{r})$, since, to express the latter, we let $\boldsymbol{c}(s) = 1$ for all $s \in \overline{S}$. The following proposition states that to solve $\mathrm{Pratio}_{\varsigma_0}(\boldsymbol{r}/\boldsymbol{c})(\boldsymbol{v})$ it suffices to solve $\mathrm{Pmp}_{\varsigma_0}(\boldsymbol{r})(\boldsymbol{v} \bullet \boldsymbol{c})$.

Proposition 3. *A finite-memory* Player $\Diamond$ *strategy* π *satisfies* $\mathrm{Pratio}_{\varsigma_0}(\boldsymbol{r}/\boldsymbol{c})(\boldsymbol{v})$ *if and only if it satisfies* $\mathrm{Pmp}_{\varsigma_0}(\boldsymbol{r})(\boldsymbol{v} \bullet \boldsymbol{c})$.

Fairness. Given a composed PA $\mathcal{M} = \|_{i \in I} M^i$, a strategy σ is *fair* if at least one action of each component $\mathcal{M}_i$ is chosen infinitely often with probability 1. We write $\mathcal{M} \models^f \varphi$ if, for all fair strategies σ, $\mathcal{M}^\sigma \models \varphi$.

Theorem 3. *Given compatible PAs* $\mathcal{M}_1$ *and* $\mathcal{M}_2$*, specifications* φ^{G_1} *and* φ^{G_2} *defined on traces of* $\mathcal{A}_{G_i} \subseteq \mathcal{A}_i$ *for* $i \in \{1, 2\}$*, then the following is sound:*

$$\frac{\mathcal{M}_1 \models^f \varphi^{G_1} \qquad \mathcal{M}_2 \models^f \varphi^{G_2}}{\mathcal{M}_1 \parallel \mathcal{M}_2 \models^f \varphi^{G_1} \wedge \varphi^{G_2}}.$$

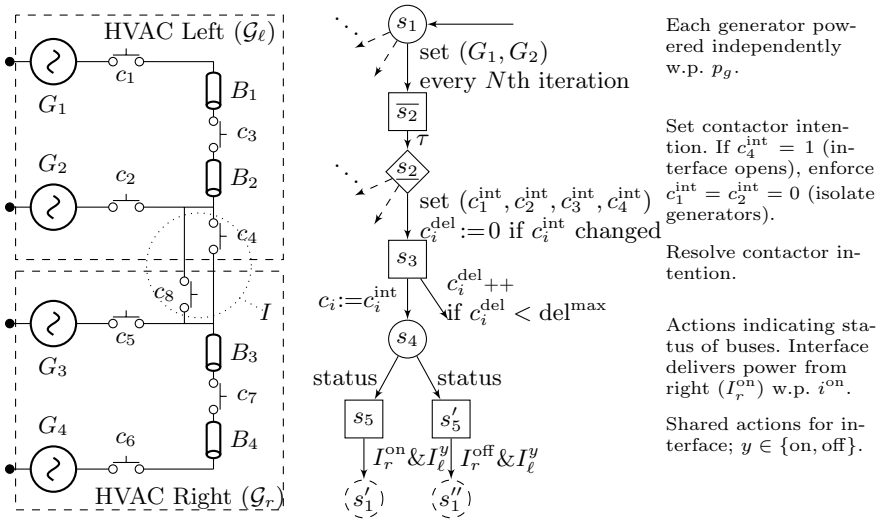
To use Theorem 3, we show that objectives using total or ratio rewards are defined on traces over some subset of actions.

Proposition 4. *If* n*-dimensional reward structures* $\boldsymbol{r}$ *and* $\boldsymbol{c}$ *are defined on actions* $\mathcal{A}_{\boldsymbol{r}}$ *and* $\mathcal{A}_{\boldsymbol{c}}$*, respectively, then objectives using ratio rewards* $\mathrm{ratio}(\boldsymbol{r}/\boldsymbol{c})$ *are defined on traces of* $\mathcal{A}_{\boldsymbol{r}} \cup \mathcal{A}_{\boldsymbol{c}}$.

Note that average rewards are not defined over traces in general, since its divisor counts the transitions, irrespective of whether the specification takes them into account. In particular, when composing systems, the additional transitions in between those originally counted skew the value of the average rewards. Moreover, τ-transitions are counted, but do not appear in the traces.

5 A Case Study: Aircraft Power Distribution

We demonstrate our synthesis methods on a case study for the control of the electrical power system of a more-electric aircraft [11], see Figure 3(a). Power is to be routed from generators to buses (and loads attached to them) by controlling the contactors (i.e. controllable switches) connecting the network nodes. Our models are based on a game-theoretic study of the same control problem in [16], where the control objective is to ensure the buses are powered, while avoiding unsafe configurations. The controllers have to take into account that contactors have delays, and the generators available in the system may be reconfigured, or even exhibit failures. We show that, by incorporating stochasticity in the models derived from the reliability statistics of the generators, controllers synthesised from ratio rewards achieve better uptimes compared to those reported in [16].



(a) Single-line diagram. (b) HVAC Left ($\mathcal{G}_\ell$).

Fig. 3: Aircraft electric power system, adapted from a Honeywell, Inc. patent [11]. The single-line diagram of the full power system (a) shows how power from the generators (G_i) can be routed to the buses (B_i) through the contactors (c_i). The left HVAC subsystem model $\mathcal{G}_\ell$ is shown in (b), and $\mathcal{G}_r$ is symmetric. I_ℓ^x and I_r^y is the interface status on the left and right side, resp., where x, y stand for either "on" or "off". One iteration of the reactive loop goes from s_1 to s_5 and starts again at s_1, potentially with some variables changed, indicated as s_1' or s_1''.

5.1 Model

The system comprises several components, each consisting of buses and generators, and we consider the high-voltage AC (HVAC) subsystem, shown in Figure 3(a), where the dashed boxes represent the components set out in [11]. These components are physically separated for reliability, and hence allow limited interaction and communication. Since the system is reactive, i.e. the aircraft is to be controlled continually, we use long-run properties to specify correctness.

The game models and control objectives in [16] are specified using LTL properties. We extend their models to stochastic games with quantitative specifications, where the contactors are controlled by Player $\Diamond$ and the contactor dynamics and the interfaces are controlled by Player $\Box$, and compose them by means of the synchronising parallel composition of [3]. The advantage of stochasticity is that the reliability specifications desired in [16] can be faithfully encoded. Further, games allow us to model truly adversarial behaviour (e.g. uncontrollable contactor dynamics), as well as nondeterministic interleaving in the composition.

Contactors, Buses and Generators. We derive the models based on the LTL description of [16]: the status of the buses and generators are kept in

Boolean variables $B_1, \ldots, B_4$ and $G_1, \ldots, G_4$ resp., and their truth value represents whether the bus or generator is powered; the contactor status is kept in Boolean variables $c_1, \ldots, c_8$, and their truth value represents if the corresponding contactor lets the current flow. For instance, if in $\mathcal{G}_\ell$ the generator G_1 is on but G_2 is off, the controller needs to switch the contactors c_1 and c_3 on, in order to power both buses B_1 and B_2. At the same time, short circuits from connecting generators to each other must be avoided, e.g. contactors c_1, c_2 and c_3 cannot be on at the same time, as this configuration connects G_1 and G_2. The contactors are, for example, solid state power controllers [14], which typically have non-negligible reaction times with respect to the times the buses should be powered. Hence, as in [16], we model that Player $\Diamond$ can only set the *intent* c_i^{int} of contactor i, and only after some delay is the contactor status c_i set to this intent. For the purposes of this demonstration, we only model a delayed turn-off time, as it is typically larger than the turn-on time (e.g. 40 ms, the turn-off time reported in [8]). Whether or not a contactor is delayed is controlled by Player $\Box$.

Interface. The components can deliver power to each other via the interface I, see Figure 3(a), which is bidirectional, i.e. power can flow both ways. The original design in [11] does not include connector c_8, and so c_4 has to ensure that no short circuits occur over the interface: if B_3 is powered, c_4 may only connect if B_2 is unpowered, and vice versa; hence, c_4 can only be on if both B_2 and B_3 are unpowered. By adding c_8, we break this cyclic dependence.

Actions shared between components model transmission of power. The actions I_r^x and I_ℓ^y for $x, y \in \{\text{on}, \text{off}\}$ model whether power is delivered via the interface from the right or left, respectively, or not. Hence, power flows from left to right via c_8, and from right to left via c_4; and we ensure via the contactors that power cannot flow in the other direction, preventing short circuits.

Reactive Loop. We model each component as an infinite loop of Player $\Box$ and Player $\Diamond$ actions. One iteration of the loop, called *time step*, represents one time unit T, and the system steps through several stages, corresponding to the states in $\mathcal{G}_\ell$ (and $\mathcal{G}_r$): in s_1 the status of the generators is set every Nth time step; in s_2 the controller sets the contactors; in s_3 the delay is chosen nondeterministically; in s_4 actions specify whether both buses are powered, and whether a failure occurs; and in s_5 information is transmitted over the interface. The τ-labelled Dirac transitions precede all Player $\Diamond$ states to enable composition [3].

Generator Assumptions. We assume that the generator status remains the same for N time steps, i.e. after 0, N, $2N$, ... steps the status may change, with the generators each powered with probability p_g, independently from each other. N and p_g can be obtained from the mean-time-to-failure of the generators. This is in contrast to [16], where, due to non-probabilistic modelling, the strongest assumption is that generators do not fail at the same time.

5.2 Specifications and Results

The main objective is to maximise uptime of the buses, while avoiding failures due to short circuits, as in [16]. Hence, the controller has to react to the gener-

Table 1: Performance statistics, for various choices of b (bus uptime), f (failure rate), i^{on} (interface uptime), and model and algorithm parameters. A minus ($-$) for i^{on} means the interface is not used. The Pareto and Strategy columns show the times for EE Pareto set computation and strategy construction, respectively.

Target			Model Params.				Algorithm Params.		Runtime [s]	
b	f	i^{on}	N	$\text{del}^{\max}$	p_g	$\lvert S\rvert$	ε	k	Pareto	Strategy
0.90	0.01	$-$	0	0	0.8	1152	0.001	20	25	0.29
0.85	0.01	$-$	3	1	0.8	15200	0.001	65	1100	2.9
0.90	0.01	$-$	3	1	0.8	15200	0.001	118	2100	2.1
0.90	0.01	0.6	0	0	0.8	2432	0.01	15	52	0.53
0.95	0.01	0.6	0	0	0.8	2432	0.01	15	49	0.46
0.90	0.01	0.6	2	1	0.8	24744	0.01	80	4300	4.80

ator status, and cannot just leave all contactors connected. The properties are specified as ratio rewards, since we are interested in the proportion of time the buses are powered. To use Theorem 3, we attach all rewards to the status actions or the synchronised actions I^x_ℓ and I^y_r. Moreover, every time step, the reward structure t attaches T to these actions to measure the progress of time.

The reward structure "buses_ℓ" (resp. "buses_r") assigns T for each time unit both buses of $\mathcal{G}_\ell$ (resp. $\mathcal{G}_r$) are powered; and the reward structure "fail_ℓ" (resp. "fail_r") assigns 1 for every time unit a short circuit occurs in $\mathcal{G}_\ell$ (resp. $\mathcal{G}_r$). Since the synchronised actions I^{on}_r and I^{on}_ℓ are taken whenever power is delivered over the interface, we attach reward structures, with the same name, assigning T whenever the corresponding action is taken. For each component $x \in \{\ell, r\}$, the objectives are to keep the uptime of the buses above b, i.e. $P^{\text{bus}}_x \equiv \text{Pratio}_{\varsigma_0}(\text{buses}_x/t)(b)$; to keep the failure rate below f, i.e. $P^{\text{safe}}_x \equiv \text{Pratio}_{\varsigma_0}(-\text{fail}_x/t)(-f)$, where minimisation is expressed using negation; and, if used, to keep the interface uptime above i^{on}, i.e. $P^{\text{int}}_x \equiv \text{Pratio}_{\varsigma_0}(I^{\text{on}}_x/t)(i^{\text{on}})$. We hence consider the specification $P^{\text{bus}}_x \wedge P^{\text{safe}}_x \wedge P^{\text{int}}_x$, for $x \in \{\ell, r\}$. Using the rule from Theorem 3 in Theorem 1 of [3], we obtain the strategy composed of the individual strategies to control the full system, satisfying $P^{\text{bus}}_\ell \wedge P^{\text{safe}}_\ell \wedge P^{\text{bus}}_r \wedge P^{\text{safe}}_r$, i.e. both components are safe and the buses are powered.

Strategy Synthesis. We implement the algorithms of this paper as an extension of our multi-objective strategy synthesis tool of [7], using a compact representation of the polyhedra $F^k_M(\top)$. Table 1 shows, for several parameter choices, the experimental results, which were obtained on a 2.8 GHz PC with 32 GB RAM. In [16], the uptime objective was encoded in LTL by requiring that buses are powered at least every Kth time step, yielding an uptime for the buses of $1/K$, which translates to an uptime of 20% (by letting $K = 5$). In contrast, using stochastic games we can utilise the statistics of the generator reliability, and obtain bus uptimes of up to 95% for generator health $p_g = 0.8$. For the models without delay, the synthesised strategies approximate memoryless deterministic strategies but when adding delay, randomisation is introduced in the memory updates. The model will be included in a forthcoming release of our tool.

6 Conclusion

We synthesise strategies for almost sure satisfaction of multi-dimensional average and ratio objectives, and demonstrate their application to assume-guarantee controller synthesis. It would be interesting to study the complexity class of the problem considered here. Satisfaction for arbitrary thresholds is subject to further research. Solutions involving an oracle computing the almost-sure winning region [9] would need to be adapted to handle our ε-approximations. Moreover, we are interested in strategies for disjunctions of satisfaction objectives.

Acknowledgements. Part of this work was sponsored by ERC AdG-246967 VERIWARE, and AFOSR grant FA9550-12-1-0302, ONR grant N000141310778.

References

1. C. Baier, C. Dubslaff, S. Klüppelholz, and L. Leuschner. Energy-utility analysis for resilient systems using probabilistic model checking. In *PETRI NETS*, vol. 8489 of LNCS, pages 20–39. Springer, 2014.
2. N. Basset, M. Kwiatkowska, U. Topcu, and C. Wiltsche. Strategy synthesis for stochastic games with multiple long-run objectives. Technical Report RR-14-10, University of Oxford, 2014.
3. N. Basset, M. Kwiatkowska, and C. Wiltsche. Compositional controller synthesis for stochastic games. In *CONCUR*, vol. 8704 of LNCS, pages 173–187. Springer, 2014.
4. T. Brázdil, V. Brozek, K. Chatterjee, V. Forejt, and A. Kucera. Two views on multiple mean-payoff objectives in Markov decision processes. *LMCS*, 10(1), 2014.
5. K. Chatterjee, M. Randour, and J.F. Raskin. Strategy synthesis for multi-dimensional quantitative objectives. *Acta Inf.*, 51(3–4):129–163, 2014.
6. T. Chen, V. Forejt, M. Kwiatkowska, A. Simaitis, and C. Wiltsche. On stochastic games with multiple objectives. In *MFCS*, vol. 8087 of LNCS, pages 266–277. Springer, 2013.
7. T. Chen, M. Kwiatkowska, A. Simaitis, and C. Wiltsche. Synthesis for multi-objective stochastic games: An application to autonomous urban driving. In *QEST*, vol. 8054 of LNCS, pages 322–337. Springer, 2013.
8. Automation Direct. Part number AD-SSR610-AC-280A, Relays and Timers, Book 2 (14.1), eRL-45, 2014.
9. H. Gimbert and F. Horn. Solving simple stochastic tail games. In *SODA*, pages 847–862. SIAM, 2010.
10. M. Kwiatkowska, G. Norman, D. Parker, and H. Qu. Compositional probabilistic verification through multi-objective model checking. *I&C*, 232:38–65, 2013.
11. R.G. Michalko. Electrical starting, generation, conversion and distribution system architecture for a more electric vehicle, 2008. US Patent 7,439,634.
12. R. Segala. *Modelling and Verification of Randomized Distributed Real Time Systems.* PhD thesis, Massachusetts Institute of Technology, 1995.
13. Lloyd S Shapley. Stochastic games. *Proc. Natl. Acad. Sci. USA*, 39(10):1095, 1953.
14. M. Sinnett. 787 no-bleed systems: saving fuel and enhancing operational efficiencies. *Aero Quarterly*, pages 6–11, 2007.
15. C. von Essen. *Quantitative Verification and Synthesis.* PhD thesis, VERIMAG, 2014.
16. H. Xu, U. Topcu, and R.M. Murray. Reactive protocols for aircraft electric power distribution. In *CDC*. IEEE, 2012.