# Strategy Synthesis for Multi-Dimensional Quantitative Objectives

Krishnendu Chatterjee[1,⋆], Mickael Randour[2,⋆⋆], and Jean-François Raskin[3,⋆⋆⋆]

[1] IST Austria (Institute of Science and Technology Austria)
[2] Institut d'Informatique, Université de Mons (UMONS), Belgium
[3] Départment d'Informatique, Université Libre de Bruxelles (U.L.B.), Belgium

**Abstract.** Multi-dimensional mean-payoff and energy games provide the mathematical foundation for the quantitative study of reactive systems, and play a central role in the emerging quantitative theory of verification and synthesis. In this work, we study the strategy synthesis problem for games with such multi-dimensional objectives along with a parity condition, a canonical way to express $\omega$-regular conditions. While in general, the winning strategies in such games may require infinite memory, for synthesis the most relevant problem is the construction of a finite-memory winning strategy (if one exists). Our main contributions are as follows. First, we show a tight exponential bound (matching upper and lower bounds) on the memory required for finite-memory winning strategies in both multi-dimensional mean-payoff and energy games along with parity objectives. This significantly improves the triple exponential upper bound for multi energy games (without parity) that could be derived from results in literature for games on VASS (vector addition systems with states). Second, we present an optimal symbolic and incremental algorithm to compute a finite-memory winning strategy (if one exists) in such games. Finally, we give a complete characterization of when finite memory of strategies can be traded off for randomness. In particular, we show that for one-dimension mean-payoff parity games, randomized memoryless strategies are as powerful as their pure finite-memory counterparts.

## 1 Introduction

Two-player games on graphs provide the mathematical foundation to study many important problems in computer science. Game-theoretic formulations have especially proved useful for synthesis [18,33,31], verification [2], refinement [29], and compatibility checking [19] of reactive systems, as well as in analysis of emptiness of automata [35].

Games played on graphs are repeated games that proceed for an infinite number of rounds. The *state* space of the graph is partitioned into player 1 states and player 2 states (player 2 is adversary to player 1). The game starts at an initial state, and if the

⋆ Author supported by Austrian Science Fund (FWF) Grant No P 23499-N23, FWF NFN Grant No S11407 (RiSE), ERC Start Grant (279307: Graph Games), Microsoft faculty fellowship.

⋆⋆ Author supported by F.R.S.-FNRS. fellowship.

⋆⋆⋆ Author supported by ERC Starting Grant (279499: inVEST).

M. Koutny and I. Ulidowski (Eds.): CONCUR 2012, LNCS 7454, pp. 115–131, 2012.

current state is a player 1 (resp. player 2) state, then player 1 (resp. player 2) chooses an outgoing *edge*. This choice is made according to a *strategy* of the player: given the sequence of visited states, a *pure* (resp. *randomized*) strategy chooses an outgoing edge (resp. probability distribution over outgoing edges). This process of choosing edges is repeated forever, and gives rise to an outcome of the game, called a *play*, that consists of the infinite sequence of states that are visited.

Traditionally, games on graphs have been studied with Boolean objectives such as reachability, liveness, $\omega$-regular conditions formalized as the canonical parity objectives, strong fairness objectives, etc [28,24,25,38,35,27]. While games with *quantitative* objectives have been studied in the game theory literature [23,39,30], their application in synthesis and other problems in verification is quite recent. The two classical quantitative objectives that are most relevant in verification and synthesis are the *mean-payoff* and *energy* objectives. In games on graphs with quantitative objectives, the game graph is equipped with a weight function that assigns integer-valued weights to every edge. For mean-payoff objectives, the goal of player 1 is to ensure that the long-run average of the weights is above a threshold. For energy objectives, the goal of player 1 is to ensure that the sum of the weights stays above 0 at all times. In applications of verification and synthesis, the quantitative objectives that typically arise are (i) multi-dimensional quantitative objectives (i.e., conjunction of several quantitative objectives), e.g., to express properties like the average response time between a grant and a request is below a given threshold $\nu_1$, and the average number of unnecessary grants is below threshold $\nu_2$; and (ii) conjunction of quantitative objectives with a Boolean objective, such as a mean-payoff parity objective that can express properties like the average response time is below a threshold along with satisfying a liveness property. In summary, the quantitative objectives can express properties related to resource requirements, performance, and robustness; multiple objectives can express the different, potentially dependent or conflicting objectives; and the Boolean objective specifies functional properties such as liveness or fairness. The game theoretic framework of multi-dimensional quantitative games and games with conjunction of quantitative and Boolean objectives has recently been shown to have many applications in verification and synthesis, such as synthesizing systems with quality guarantee [4], synthesizing robust systems [5], performance aware synthesis of concurrent data structure [10], analyzing permissivity in games and synthesis [8], simulation between quantitative automata [14], generalizing Boolean simulation to quantitative simulation distance [11], etc. Moreover, multi-dimensional energy games are equivalent to a decidable class of games on VASS (vector addition systems with states) that are the model to verify games over multi-counter systems and Petri nets [9].

In literature, there are many recent works on the theoretical analysis of multi-dimensional quantitative games, such as, mean-payoff parity games [16,8], energy-parity games [13], multi-dimensional energy games [15], and multi-dimensional mean-payoff games [15,37]. Most of these works focus on establishing the computational complexity of the problem of deciding if player 1 *has* a *winning* strategy. From the perspective of synthesis and other related problems in verification, the most important problem is to obtain a witness *finite-memory* winning strategy (if one exists). The winning strategy in the game corresponds to the desired controller for (or implementation of) the system in synthesis, and for implementability a finite-memory strategy is essential. In this work we consider the problem of finite-memory strategy synthesis in multi-dimensional quantitative games in conjunction with parity objectives, and the problem of existence of memory-efficient randomized strategies for such games. These are the core and foundational problems in the emerging theory of quantitative verification and synthesis.

**Our Contributions.** In this work, we study for the first time multi-dimensional energy and mean-payoff objectives in conjunction with parity objectives. Conjunction of parity objectives with multi-dimensional quantitative objectives has not been considered before. Since we consider the synthesis of finite-memory strategies, it follows from the results of [15] that both the problems (multi-dimensional energy with parity and multi-dimensional mean-payoff with parity) are equivalent. Our main results for finite-memory strategy synthesis for multi-dimensional energy parity games are as follows. **(**$i$**)** **Optimal memory bounds**. We first show that memory of exponential size is sufficient in multi-dimensional energy parity games. Our result is a significant improvement over the result that can be obtained naively from the results known in literature that yields a triple exponential bound, even in the case of multi-dimensional energy games without parity. Second, we show a matching lower bound by presenting a family of game graphs where exponential memory is necessary in multi-dimensional energy games (without parity), even when all the transition weights belong to $\{-1, 0, +1\}$. Thus we establish *optimal memory bounds* for the finite-memory strategy synthesis problem. **(**$ii$**)** **Symbolic and incremental algorithm.** We present a *symbolic* algorithm (in the sense of [21], i.e., using a compact antichain representation of sets by their minimal elements) to compute a finite-memory winning strategy, if one exists, for multi-dimensional energy parity games. Our algorithm is parameterized by the range of energy levels to consider during its execution. So, we can use it in an *incremental approach*: first, we search for finite-memory winning strategies with a small range, and increment the range only when necessary. We also establish a bound on the maximal range to consider which ensures completeness of the incremental approach. In the worst case the algorithm requires exponential time. Since exponential size memory is required (and also the decision problem is coNP-complete [15]), the worst case exponential bound can be considered as *optimal*. Moreover, as our algorithm is symbolic and incremental, in most relevant problems in practice, it is expected to be efficient. We also consider when the (pure) finite-memory strategies can be traded off for conceptually much simpler randomized strategies. **(**$iii$**)** **Randomized strategies.** We show that for energy objectives randomization is not helpful (as energy objectives are similar in spirit with safety objectives), even with only one player, neither it is for two-player multi-dimensional mean-payoff objectives. However, randomized memoryless strategies suffice for one-player multi-dimensional mean-payoff parity games. For the important special case of mean-payoff parity objectives (conjunction of a single mean-payoff and parity objectives), we show that in games, finite-memory strategies can be traded off for randomized memoryless strategies. An extended version of this work, including proofs, can be found in [17].

**Related Works.** Games with a single mean-payoff objective have been studied in [23,39], and games with a single energy objective in [12]; their equivalence was established in [7]. One-dimensional mean-payoff parity games problem has been studied in [16]: an exponential algorithm was given to decide if there exists a winning strategy (which in general was shown to require infinite memory); and an improved algorithm was presented in [8]. One-dimensional energy parity games problem has been studied in [13]: it was shown that deciding the existence of a winning strategy is in NP $\cap$ coNP, and an exponential algorithm was given. It was also shown in [13] that, for one-dimensional energy parity objectives, finite-memory strategies with exponential memory are sufficient, and the decision problem for mean-payoff parity objective can be reduced to energy parity objective. Games on VASS with several different winning objectives have been studied in [9], and from the results of [9] it follows that in multi-dimensional energy games, winning strategies with finite memory are sufficient (and a triple exponential bound on memory can be derived from the results). The complexity of multi-dimensional energy and mean-payoff games was studied in [15,37]. It was shown in [15] that in general, winning strategies in multi-dimensional mean-payoff games require infinite memory, whereas for multi-dimensional energy games, finite-memory strategies are sufficient. Moreover, for finite-memory strategies, the multi-dimensional mean-payoff and energy games coincide, and optimal computational complexity for deciding the existence of a winning strategy was established as coNP-complete [15,37]. Multi-dimensional mean-payoff games with infinite-memory strategies were studied in [37], and optimal computational complexity results were established. Various decision problems over multi-dimensional energy games were studied in [26].

## 2 Preliminaries

We consider two-player game structures and denote the two *players* by $\mathcal{P}_1$ and $\mathcal{P}_2$.

**Multi-Weighted Two-Player Game Structures.** A *multi-weighted two-player game structure* is a tuple $G = (S_1, S_2, s_{init}, E, k, w)$ where (i) $S_1$ and $S_2$ resp. denote the finite sets of *states* belonging to $\mathcal{P}_1$ and $\mathcal{P}_2$, with $S_1 \cap S_2 = \emptyset$; (ii) $s_{init} \in S = S_1 \cup S_2$ is the initial state; (iii) $E \subseteq S \times S$ is the set of *edges* s.t. for all $s \in S$, there exists $s' \in S$ s.t. $(s, s') \in E$; (iv) $k \in \mathbb{N}$ is the *dimension* of the weight vectors; and (v) $w\colon E \to \mathbb{Z}^k$ is the multi-weight labeling function. The game structure $G$ is *one-player* if $S_2 = \emptyset$. A *play* in $G$ is an infinite sequence of states $\pi = s_0 s_1 s_2 \ldots$ s.t. $s_0 = s_{init}$ and for all $i \geq 0$, we have $(s_i, s_{i+1}) \in E$. The *prefix* up to the $n$-th state of play $\pi = s_0 s_1 \ldots s_n \ldots$ is the finite sequence $\pi(n) = s_0 s_1 \ldots s_n$. Let $\mathsf{First}(\pi(n))$ and $\mathsf{Last}(\pi(n))$ resp. denote $s_0$ and $s_n$, the first and last states of $\pi(n)$. A prefix $\pi(n)$ belongs to $\mathcal{P}_i$, $i \in \{1, 2\}$, if $\mathsf{Last}(\pi(n)) \in S_i$. The set of plays of $G$ is denoted by $\mathsf{Plays}(G)$ and the corresponding set of prefixes is denoted by $\mathsf{Prefs}(G)$. The set of prefixes that belong to $\mathcal{P}_i$ is denoted by $\mathsf{Prefs}_i(G)$. The *energy level vector* of a sequence of states $\rho = s_0 s_1 \ldots s_n$ s.t. for all $i \geq 0$, we have $(s_i, s_{i+1}) \in E$, is $\mathsf{EL}(\rho) = \sum_{i=0}^{i=n-1} w(s_i, s_{i+1})$ and the *mean-payoff vector* of a play $\pi = s_0 s_1 \ldots$ is $\mathsf{MP}(\pi) = \liminf_{n \to \infty} \frac{1}{n} \mathsf{EL}(\pi(n))$.

**Parity.** A game structure $G$ is extended with a priority function $p\colon S \to \mathbb{N}$ to $G_p = (S_1, S_2, s_{init}, E, k, w, p)$. Given a play $\pi = s_0 s_1 s_2 \ldots$, let $\mathsf{Inf}(\pi) = \{s \in S \mid \forall m \geq 0, \exists\, n > m \text{ s.t. } s_n = s\}$ denote the set of states that appear infinitely often along $\pi$. The *parity* of a play $\pi$ is defined as $\mathsf{Par}(\pi) = \min\{p(s) \mid s \in \mathsf{Inf}(\pi)\}$. In the following definitions, we denote any game by $G_p$ with no loss of generality.

**Strategies.** Given a finite set $A$, a *probability distribution* on $A$ is a function $p\colon A \mapsto [0,1]$ s.t. $\sum_{a \in A} p(a) = 1$. We denote the set of probability distributions on $A$ by $\mathcal{D}(A)$. A *pure strategy* for $\mathcal{P}_i$, $i \in \{1,2\}$, in $G_p$ is a function $\lambda_i\colon \mathsf{Prefs}_i(G_p) \to S$ s.t. for all $\rho \in \mathsf{Prefs}_i(G_p)$, we have $(\mathsf{Last}(\rho), \lambda_i(\rho)) \in E$. A *(behavioral) randomized strategy* is a function $\lambda_i\colon \mathsf{Prefs}_i(G_p) \to \mathcal{D}(S)$ s.t. for all $\rho \in \mathsf{Prefs}_i(G_p)$, we have $\{(\mathsf{Last}(\rho), s) \mid s \in S, \lambda_i(\rho)(s) > 0\} \subseteq E$. A pure strategy $\lambda_i$ for $\mathcal{P}_i$ has *finite-memory* if it can be encoded by a deterministic Moore machine $(M, m_0, \alpha_u, \alpha_n)$ where $M$ is a finite set of states (the memory of the strategy), $m_0 \in M$ is the initial memory state, $\alpha_u\colon M \times S \to M$ is an update function, and $\alpha_n\colon M \times S_i \to S$ is the next-action function. If the game is in $s \in S_i$ and $m \in M$ is the current memory value, then the strategy chooses $s' = \alpha_n(m,s)$ as the next state of the game. When the game leaves a state $s \in S$, the memory is updated to $\alpha_u(m,s)$. Formally, $\langle M, m_0, \alpha_u, \alpha_n \rangle$ defines the strategy $\lambda_i$ s.t. $\lambda_i(\rho \cdot s) = \alpha_n(\hat{\alpha}_u(m_0, \rho), s)$ for all $\rho \in S^*$ and $s \in S_i$, where $\hat{\alpha}_u$ extends $\alpha_u$ to sequences of states as expected. A pure strategy is *memoryless* if $|M| = 1$, i.e., it does not depend on history but only on the current state of the game. Similar definitions hold for finite-memory randomized strategies, s.t. the next-action function $\alpha_n$ is randomized, while the update function $\alpha_u$ remains deterministic. We resp. denote by $\Lambda_i, \Lambda_i^{PF}, \Lambda_i^{PM}, \Lambda_i^{RM}$ the sets of general (i.e., possibly randomized and infinite-memory), pure finite-memory, pure memoryless and randomized memoryless strategies for player $\mathcal{P}_i$.

Given a prefix $\rho \in \mathsf{Prefs}_i(G_p)$ belonging to player $\mathcal{P}_i$, and a strategy $\lambda_i \in \Lambda_i$ of this player, we define the *support* of the probability distribution defined by $\lambda_i$ as $\mathsf{Supp}_{\lambda_i}(\rho) = \{s \in S \mid \lambda_i(\rho)(s) > 0\}$, with $\lambda_i(\rho)(s) = 1$ if $\lambda_i$ is pure and $\lambda_i(\rho) = s$. A play $\pi$ is said to be *consistent* with a strategy $\lambda_i$ of $\mathcal{P}_i$ if for all $n \geq 0$ s.t. $\mathsf{Last}(\pi(n)) \in S_i$, we have $\mathsf{Last}(\pi(n+1)) \in \mathsf{Supp}_{\lambda_i}(\pi(n))$. Given two strategies, $\lambda_1$ for $\mathcal{P}_1$ and $\lambda_2$ for $\mathcal{P}_2$, we define $\mathsf{Outcome}_{G_p}(\lambda_1, \lambda_2) = \{\pi \in \mathsf{Plays}(G_p) \mid \pi \text{ is consistent with } \lambda_1 \text{ and } \lambda_2\}$, the set of possible *outcomes* of the game. Note that if both strategies $\lambda_1$ and $\lambda_2$ are pure, we obtain a unique play $\pi = s_0 s_1 s_2 \ldots$ s.t. for all $j \geq 0$, $i \in \{1,2\}$, if $s_j \in S_i$, then we have $s_{j+1} = \lambda_i(s_j)$.

Given the initial state $s_{init}$ and strategies for both players $\lambda_1 \in \Lambda_1$, $\lambda_2 \in \Lambda_2$, we obtain a Markov chain. Thus, every *event* $\mathcal{A} \subseteq \mathsf{Plays}(G_p)$, a measurable set of plays, has a uniquely defined probability [36]. We denote by $\mathbb{P}_{s_{init}}^{\lambda_1,\lambda_2}(\mathcal{A})$ the probability that a play belongs to $\mathcal{A}$ when the game starts in $s_{init}$ and is played consistently with $\lambda_1$ and $\lambda_2$. We use the same notions for prefixes by naturally extending them to their infinite counterparts.

**Objectives.** An *objective* for $\mathcal{P}_1$ in $G_p$ is a set of plays $\phi \subseteq \mathsf{Plays}(G_p)$. We consider several kinds of objectives:

- *Multi Energy objectives*. Given an initial energy vector $v_0 \in \mathbb{N}^k$, the objective $\mathsf{PosEnergy}_{G_p}(v_0) = \left\{\pi \in \mathsf{Plays}(G_p) \mid \forall\, n \geq 0 : v_0 + \mathsf{EL}(\pi(n)) \in \mathbb{N}^k\right\}$ requires that the energy level in all dimensions stays positive at all times.
- *Multi Mean-payoff objectives*. Given a threshold vector $v \in \mathbb{Q}^k$, the objective $\mathsf{MeanPayoff}_{G_p}(v) = \{\pi \in \mathsf{Plays}(G_p) \mid \mathsf{MP}(\pi) \geq v\}$ requires that for all dimension $j$, the mean-payoff on this dimension is at least $v(j)$.
- *Parity objectives*. Objective $\mathsf{Parity}_{G_p} = \{\pi \in \mathsf{Plays}(G_p) \mid \mathsf{Par}(\pi) \bmod 2 = 0\}$ requires that the minimum priority visited infinitely often be even. When the set of

priorities is restricted to $\{0, 1\}$, we have a *Büchi objective*. Note that every multi-weighted game structure $G$ without parity can trivially be extended to $G_p$ with $p\colon S \to \{0\}$.

- *Combined objectives*. Parity can naturally be combined with multi mean-payoff and multi energy objectives, resp. yielding $\mathsf{MeanPayoff}_{G_p}(v) \cap \mathsf{Parity}_{G_p}$ and $\mathsf{PosEnergy}_{G_p}(v_0) \cap \mathsf{Parity}_{G_p}$.

**Sure and Almost-Sure Semantics.** A strategy $\lambda_1$ for $\mathcal{P}_1$ is *surely winning* for an objective $\phi$ in $G_p$ if for all plays $\pi \in \mathsf{Plays}(G_p)$ that are consistent with $\lambda_1$, we have $\pi \in \phi$. When at least one of the players plays a randomized strategy, the notion of sure winning in general is too restrictive and inadequate, as the set of consistent plays that do not belong to $\phi$ may have zero probability measure. Therefore, we use the concept of *almost-surely winning*. Given a measurable objective $\phi \subseteq \mathsf{Plays}(G_p)$, a strategy $\lambda_1$ for $\mathcal{P}_1$ is *almost-surely winning* if for all $\lambda_2 \in \Lambda_2$, we have $\mathbb{P}^{\lambda_1,\lambda_2}_{s_{init}}(\phi) = 1$.

**Strategy Synthesis Problem.** For multi energy parity games, the problem is to synthesize a finite initial credit $v_0 \in \mathbb{N}^k$ and a pure *finite-memory* strategy $\lambda_1^{pf} \in \Lambda_1^{PF}$ that is surely winning for $\mathcal{P}_1$ in $G_p$ for the objective $\mathsf{PosEnergy}_{G_p}(v_0) \cap \mathsf{Parity}_{G_p}$, *if one exists*. So, the initial credit is not fixed, but is part of the strategy to synthesize. For multi mean-payoff games, given a threshold $v \in \mathbb{Q}^k$, the problem is to synthesize a pure *finite-memory* strategy $\lambda_1^{pf} \in \Lambda_1^{PF}$ that is surely winning for $\mathcal{P}_1$ in $G_p$ for the objective $\mathsf{MeanPayoff}_{G_p}(v) \cap \mathsf{Parity}_{G_p}$, *if one exists*. Note that multi energy and multi mean-payoff games are equivalent for finite-memory strategies, while in general, infinite memory may be necessary for the latter [15].

**Trading Finite Memory for Randomness.** We study when finite memory can be traded for randomization. The question is: given a strategy $\lambda_1^{pf} \in \Lambda_1^{PF}$ which ensures surely winning of some objective $\phi$, does there exist a strategy $\lambda_1^{rm} \in \Lambda_1^{RM}$ which ensures almost-surely winning for the same objective $\phi$?

## 3 Optimal Memory Bounds

In this section, we establish optimal memory bounds for pure finite-memory winning strategies on multi-dimensional energy parity games (MEPGs). Also, as a corollary, we obtain results for pure finite-memory winning strategies on multi-dimensional mean-payoff parity games (MMPPGs). We show that single exponential memory is both sufficient and necessary for winning strategies. Additionally, we show how the parity condition in a MEPG can be removed by adding additional energy dimensions.

**Multi Energy Parity Games.** A sample game is depicted on Fig. 1. The key point in the upper bound proof on memory is to understand that for $\mathcal{P}_1$ to win a multi energy parity game, he must be able to force cycles whose energy level is positive in all dimensions and whose minimal parity is even. As stated in the next lemma, finite-memory strategies are sufficient for multi energy parity games for both players.

**Lemma 1 (Extension of [15, Lemma 2 and 3]).** *If $\mathcal{P}_1$ wins a multi energy parity game, then he has a pure finite-memory winning strategy. If $\mathcal{P}_2$ wins a multi energy parity game, then he has a pure memoryless winning strategy.*

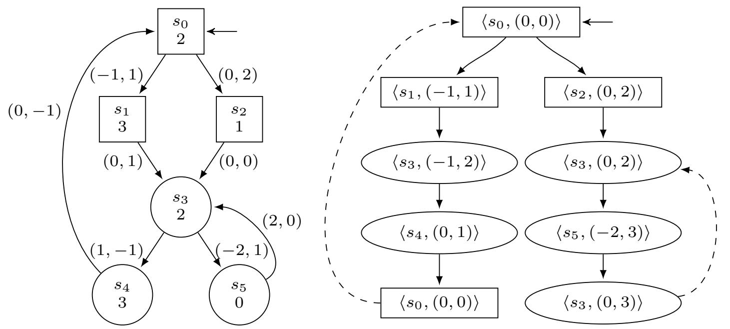


**Fig. 1.** Two-dimensional energy parity game and epSCT representing an arbitrary finite-memory winning strategy. Circle states belong to $\mathcal{P}_1$, square states to $\mathcal{P}_2$.

By Lemma 1, we know that w.l.o.g. both players can be restricted to play pure finite memory strategies. The property on the cycles can then be formalized as follows.

**Lemma 2.** *Let $G_p = (S_1, S_2, s_{init}, E, k, w, p)$ be a multi energy parity game. Let $\lambda_1^{pf} \in \Lambda_1^{PF}$ be a winning strategy of $\mathcal{P}_1$ for initial credit $v_0 \in \mathbb{N}^k$. Then, for all $\lambda_2^{pm} \in \Lambda_2^{PM}$, the outcome is a regular play $\pi = \rho \cdot (\eta_\infty)^\omega$, with $\rho \in \textsf{Prefs}(G), \eta_\infty \in S^+$, s.t. $\textsf{EL}(\eta_\infty) \geq 0$ and $\textsf{Par}(\pi) = \min\{p(s) \mid s \in \eta_\infty\}$ is even.*

With the notion of regular play of Lemma 2, we generalize the notion of *self-covering path* to include the parity condition. We show here that, if such a path exists, then the lengths of its cycle and the prefix needed to reach it can be bounded. Bounds on the strategy follow. In [32], Rackoff showed how to bound the length of self-covering paths in *Vector Addition Systems* (VAS). This work was extended to Vector Addition Systems with States (VASS) by Rosier and Yen [34]. Recently, Brázdil *et al.* introduced reachability games on VASS and the notion of *self-covering trees* [9]. Their Zero-safety problem with $\omega$ initial marking is equivalent to multi energy games with weights in $\{-1, 0, 1\}$, and without the parity condition. They showed that if winning strategies exist for $\mathcal{P}_1$, then some of them can be represented as *self-covering trees* of bounded depth. Trees have to be considered instead of paths, as in a game setting all the possible choices of the adversary ($\mathcal{P}_2$) must be considered. Here, we extend the notion of self-covering trees to *even-parity self-covering trees*, in order to handle parity objectives.

**Even-Parity Self-covering Tree.** An *even-parity self-covering tree* (epSCT) for $s \in S$ is a finite tree $T = (Q, R)$, where $Q$ is the set of nodes, $\Theta\colon Q \mapsto S \times \mathbb{Z}^k$ is a labeling function and $R \subset Q \times Q$ is the set of edges, s.t.

- The root of $T$ is labeled $\langle s, (0, \ldots, 0)\rangle$.
- If $\varsigma \in Q$ is not a leaf, then let $\Theta(\varsigma) = \langle t, u\rangle$, $t \in S$, $u \in \mathbb{Z}^k$, s.t.
  - if $t \in S_1$, then $\varsigma$ has a unique child $\vartheta$ s.t. $\Theta(\vartheta) = \langle t', u'\rangle$, $(t, t') \in E$ and $u' = u + w(t, t')$;
  - if $t \in S_2$, then there is a bijection between children of $\varsigma$ and edges of the game leaving $t$, s.t. for each successor $t' \in S$ of $t$ in the game, there is one child $\vartheta$ of $\varsigma$ s.t. $\Theta(\vartheta) = \langle t', u'\rangle$, $u' = u + w(t, t')$.

– If $\varsigma$ is a leaf, then let $\Theta(\varsigma) = \langle t, u\rangle$ s.t. there is some ancestor $\vartheta$ of $\varsigma$ in $T$ s.t. $\Theta(\vartheta) = \langle t, u'\rangle$, with $u' \leq u$, and the downward path from $\vartheta$ to $\varsigma$, denoted by $\vartheta \rightsquigarrow \varsigma$, has minimal priority even. We say that $\vartheta$ is an *even-descendance energy ancestor* of $\varsigma$.

Intuitively, each path from root to leaf is a self-covering path of even parity in the game graph so that plays unfolding according to such a tree correspond to winning plays of Lemma 2. Thus, the epSCT fixes how $\mathcal{P}_1$ should react to actions of $\mathcal{P}_2$ in order to win the MEPG (Fig. 1). Note that as the tree is finite, one can take the largest negative number that appears on a node in each dimension to compute an initial credit for which there is a winning strategy (i.e., the one described by the tree). In particular, let $W$ denote the maximal absolute weight appearing on an edge in $G_p$. Then, for an epSCT $T$ of depth $l$, it is straightforward to see that the maximal initial credit required is at most $l \cdot W$ as the maximal decrease at each level of the tree is bounded by $W$. We suppose $W > 0$ as otherwise, any strategy of $\mathcal{P}_1$ is winning for the energy objective, for any initial credit vector $v_0 \in \mathbb{N}^k$.

Let us explicitely state how $\mathcal{P}_1$ can deploy a strategy $\lambda_1^T \in \Lambda_1^{PF}$ based on an epSCT $T = (Q, R)$. We refer to such a strategy as an *epSCT strategy*. It consists in following a path in the tree $T$, moving a pebble from node to node and playing in the game depending on edges taken by this pebble. Each time a node $\varsigma$ s.t. $\Theta(\varsigma) = \langle t, u\rangle$ is encountered, we do the following.

– If $\varsigma$ is a leaf, the pebble directly goes up to its oldest even-descendance energy ancestor $\vartheta$. By oldest we mean the first encountered when going down in the tree from the root. Note that this choice is arbitrary, in a effort to ease following proof formulations, as any one would suit.
– Otherwise, if $\varsigma$ is not a leaf,
  - if $t \in S_2$ and $\mathcal{P}_2$ plays state $t' \in S$, the pebble is moved along the edge going to the only child $\vartheta$ of $\varsigma$ s.t. $\Theta(\vartheta) = \langle t', u'\rangle$, $u' = u + w(t, t')$;
  - if $t \in S_1$, the pebble moves to $\vartheta$, $\Theta(\vartheta) = \langle t', u'\rangle$, the only child of $\varsigma$, and $\mathcal{P}_1$ strategy is to choose the state $t'$ in the game.

If such an epSCT $T$ of depth $l$ exists for a game $G_p$, then $\mathcal{P}_1$ can play the strategy $\lambda_1^T \in \Lambda_1^{PF}$ to win the game with initial credit bounded by $l \cdot W$.

**Bounding the Depth of epSCTs.** Consider a multi energy game *without* parity. Then, the priority condition on downward paths from ancestor to leaf is not needed and self-covering trees (i.e., epSCTs without the condition on priorities) suffice to describe winning strategies. One can bound the size of SCTs using results on the size of solutions for linear diophantine equations (i.e., with integer variables) [6]. In particular, recent work on reachability games over VASS with weights $\{-1, 0, 1\}$, Lemma 7 of [9], states that if $\mathcal{P}_1$ has a winning strategy on a VASS, then he can exhibit one that can be described as a SCT whose *depth* is at most $l = 2^{(d-1)\cdot|S|} \cdot (|S| + 1)^{c\cdot k^2}$, where $c$ is a constant independent of the considered VASS and $d$ its branching degree (i.e., the highest number of outgoing edges on any state). Naive use of this bound for multi energy games with arbitrary integer weights would induce a *triple* exponential bound for memory. Indeed, recall that $W$ denotes the maximal absolute weight that appears in a game

$G_p = (S_1, S_2, s_{init}, E, k, w, p)$. A straightforward translation of a game with arbitrary weights into an equivalent game that uses only weights in $\{-1, 0, 1\}$ induces a blow-up by $W$ in the size of the state space, and thus an exponential blow-up by $W$ in the depth of the tree, which becomes doubly exponential as we have

$$l = 2^{(d-1)\cdot W\cdot|S|} \cdot (W \cdot |S| + 1)^{c\cdot k^2} = 2^{(d-1)\cdot 2^V\cdot|S|} \cdot (W \cdot |S| + 1)^{c\cdot k^2},$$

where $V$ denotes the number of bits used by the encoding of $W$. Moreover, the width of the tree increases as $d^l$, i.e., it increases exponentially with the depth. So straight application of previous results provides an overall tree of triple exponential size. In this paper we improve this bound and prove a single exponential upper bound, even for multi energy *parity* games. We proceed in two steps, first studying the depth of the epSCT, and then showing how to compress the tree into a *directed acyclic graph* (DAG) of *single* exponential size.

**Lemma 3.** *Let $G_p = (S_1, S_2, s_{init}, E, k, w, p)$ be a multi energy parity game s.t. $W$ is the maximal absolute weight appearing on an edge and* d *the branching degree of $G_p$. Suppose there exists a finite-memory winning strategy for $\mathcal{P}_1$. Then there is an even-parity self-covering tree for $s_{init}$ of depth at most $l = 2^{(d-1)\cdot|S|} \cdot (W \cdot |S| + 1)^{c\cdot k^2}$, where* c *is a constant independent of $G_p$.*

Lemma 3 eliminates the exponential blow-up in depth induced by a naive coding of arbitrary weights into $\{-1, 0, 1\}$ weights, and implies an overall doubly exponential upper bound. Our proof is a generalization of [9, Lemma 7], using a more refined analysis to handle both *parity* and *arbitrary integer weights*. The idea is the following. First, consider the one-player case. The epSCT is reduced to a path. By Lemma 2, it is composed of a finite prefix, followed by an infinitely repeated sequence of positive energy level and even minimal priority. The point is to bound the length of such a sequence by eliminating cycles that are not needed for energy or parity. Second, to extend the result to two-player games, we use an induction on the number of choices available for $\mathcal{P}_2$ in a given state. Intuitively, we show that if $\mathcal{P}_1$ can win with an epSCT $T_A$ when $\mathcal{P}_2$ plays edges from a set $A$ in a state $s$, and if he can also win with an epSCT $T_B$ when $\mathcal{P}_2$ plays edges from a set $B$, then he can win when $\mathcal{P}_2$ chooses edges from both $A$ and $B$, with an epSCT whose depth is bounded by the sum of depths of $T_A$ and $T_B$.

**From Multi Energy Parity Games to Multi Energy Games.** Let $G_p$ be a MEPG and assume that $\mathcal{P}_1$ has a winning strategy in that game. By Lemma 3, there exists an epSCT whose depth is bounded by $l$. As a direct consequence of that bounded depth, we have that $\mathcal{P}_1$, by playing the strategy prescribed by the epSCT, enforces a stronger objective than the parity objective. Namely, this strategy ensures to "never visit more than $l$ states of odd priorities before seeing a smaller even priority" (which is a safety objective). Then, the parity condition can be transformed into additional energy dimensions.

While our transformation shares ideas with the classical transformation of parity objectives into safety objectives, first proposed in [3] (see also [22, Lemma 6.4]), it is technically different because energy levels cannot be reset (as it would be required by those classical constructions). The reduction is as follows. For each odd priority, we add one dimension. The energy level in this dimension is decreased by 1 each time this odd priority is visited, and it is increased by $l$ each time a smaller even priority is visited. If $\mathcal{P}_1$ is able to maintain the energy level positive for all dimensions (for a given initial energy level), then he is clearly winning the original parity objective; on the other hand, an epSCT strategy that wins the original objective also wins the new game.

**Lemma 4.** *Let $G_p = (S_1, S_2, s_{init}, E, k, w, p)$ be a multi energy parity game with priorities in $\{0, 1, \ldots, 2 \cdot m\}$, s.t. $W$ is the maximal absolute weight appearing on an edge. Then we can construct a multi energy game G with the same set of states, $(k+m)$ dimensions and a maximal absolute weight bounded by $l$, as defined by Lemma 3, s.t. $\mathcal{P}_1$ has a winning strategy in G iff he has one in $G_p$.*

**Bounding the Width.** Thanks to Lemma 4, we continue with multi energy games without parity. In order to bound the overall size of memory for winning strategies, we consider the width of self-covering trees. The following lemma states that SCTs, whose width is at most doubly exponential by application of Lemma 3, can be compressed into *directed acyclic graphs* (DAGs) of single exponential width. Thus we eliminate the second exponential blow-up and give an overall single exponential bound for memory of winning strategies.

**Lemma 5.** *Let $G = (S_1, S_2, s_{init}, E, k, w)$ be a multi energy game s.t. $W$ is the maximal absolute weight appearing on an edge and $d$ the branching degree of G. Suppose there exists a finite-memory winning strategy for $\mathcal{P}_1$. Then, there exists $\lambda_1^D \in \Lambda_1^{PF}$ a winning strategy for $\mathcal{P}_1$ described by a DAG D of depth at most $l = 2^{(d-1)\cdot|S|} \cdot (W \cdot |S| + 1)^{c \cdot k^2}$ and width at most $L = |S| \cdot (2 \cdot l \cdot W + 1)^k$, where $c$ is a constant independent of G. Thus the overall memory needed to win this game is bounded by the single exponential $l \cdot L$.*

The sketch of this proof is the following. By Lemma 3, we know that there exists a tree $T$, and thus a DAG, that satisfies the bound on depth. We construct a finite sequence of DAGs, whose first element is $T$, so that (1) each DAG describes a winning strategy for the same initial credit, (2) each DAG has the same depth, and (3) the last DAG of the sequence has its width bounded by $|S| \cdot (2 \cdot l \cdot W + 1)^k$. This sequence $D_0 = T, D_1, D_2, \ldots, D_n$ is built by merging nodes on the same level of the initial tree depending on their labels, level by level. The key idea of this procedure is that what actually matters for $\mathcal{P}_1$ is only the current energy level, which is encoded in node labels in the self-covering tree $T$. Therefore, we merge nodes with identical states and energy levels: since $\mathcal{P}_1$ can essentially play the same strategy in both nodes, we only keep one of their subtrees.

**Lower Bound.** In the next lemma, we show that the upper bound is tight in the sense that there exist families of games which require exponential memory (in the number of dimensions), even for the simpler case of multi energy objectives without parity and weights in $\{-1, 0, 1\}$ (Fig. 2).

**Lemma 6.** *There exists a family of multi energy games $(G(K))_{K \geq 1,} = (S_1, S_2, s_{init}, E, k = 2 \cdot K, w\colon E \rightarrow \{-1, 0, 1\})$ s.t. for any initial credit, $\mathcal{P}_1$ needs exponential memory to win.*

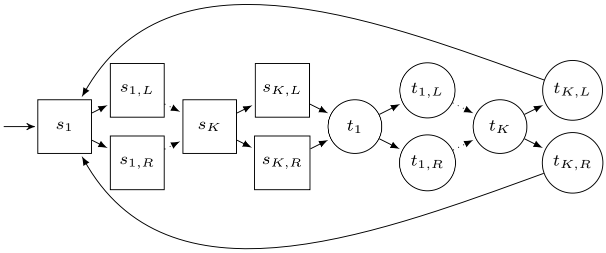


**Fig. 2.** Family of games requiring exponential memory: $\forall 1 \leq i \leq K$, $\forall 1 \leq j \leq k$, $w((s_i, s_{i,L}))(j) = 1$ if $j = 2 \cdot i - 1$, $= -1$ if $j = 2 \cdot i$, and $= 0$ otherwise; $w((s_i, s_{i,L})) = -w((s_i, s_{i,R})) = w((t_i, t_{i,L})) = -w((t_i, t_{i,R}))$; $w((\circ, s_i)) = w((\circ, t_i)) = (0, \ldots, 0)$.

The idea is the following: in the example of Fig. 2, if $\mathcal{P}_1$ does not remember the exact choices of $\mathcal{P}_2$ (which requires an exponential size Moore machine), there will exist some sequence of choices of $\mathcal{P}_2$ s.t. $\mathcal{P}_1$ cannot counteract a decrease in energy. Thus, by playing this sequence long enough, $\mathcal{P}_2$ can force $\mathcal{P}_1$ to lose, whatever his initial credit is.

We summarize our results in Theorem 1.

**Theorem 1 (Optimal memory bounds).** *The following assertions hold: (1) In multi energy parity games, if there exists a winning strategy, then there exists a finite-memory winning strategy. (2) In multi energy parity and multi mean-payoff games, if there exists a finite-memory winning strategy, then there exists a winning strategy with at most exponential memory. (3) There exists a family of multi energy games (without parity) with weights in* $\{-1, 0, 1\}$ *where all winning strategies require at least exponential memory.*

## 4 Symbolic Synthesis Algorithm

We now present a *symbolic*, *incremental* and *optimal* algorithm to synthesize a finite-memory winning strategy in a MEG.[1] This algorithm outputs a (set of) winning initial credit(s) and a derived finite-memory winning strategy (if one exists) which is exponential in the worst-case. Its running time is at most exponential. So our symbolic algorithm can be considered (worst-case) optimal in the light of the results of previous section.

This algorithm computes the greatest fixed point of a monotone operator that defines the sets of winning initial (vectors of) credits for each state of the game. As those sets are upward-closed, they are symbolically represented by their minimal elements. To ensure convergence, the algorithm considers only credits that are below some *threshold*, noted $\mathbb{C}$. This is without giving up completeness because, as we show below, for a game $G = (S_1, S_2, s_{init}, E, k, w)$, it is sufficient to take the value $2 \cdot l \cdot W$ for $\mathbb{C}$, where $l$ is the bound on the depth on epSCT obtained in Lemma 3 and $W$ is the largest absolute

[1] Note that the symbolic algorithm can be applied to MEPGs and MMPPGs after removal of the parity condition by applying the construction of Lemma 4.

value of weights used in the game. We also show how to extract a finite state Moore machine from this set of minimal winning initial credits and how to obtain an *incremental* algorithm by increasing values for the threshold $\mathbb{C}$ starting from small values.

**A Controllable Predecessor Operator.** Let $G = (S_1, S_2, s_{init}, E, k, w)$ be a MEG, $\mathbb{C} \in \mathbb{N}$ be a constant, and $U(\mathbb{C})$ be the set $(S_1 \cup S_2) \times \{0, 1, \dots, \mathbb{C}\}^k$. Let $\mathcal{U}(\mathbb{C}) = 2^{U(\mathbb{C})}$, i.e., the powerset of $U(\mathbb{C})$, and the operator $\mathsf{Cpre}_{\mathbb{C}} \colon \mathcal{U}(\mathbb{C}) \to \mathcal{U}(\mathbb{C})$ be defined as follows:

$$\mathcal{E}(V) = \{(s_1, e_1) \in U(\mathbb{C}) \mid s_1 \in S_1 \wedge \exists (s_1, s) \in E, \exists (s, e_2) \in V : e_2 \leq e_1 + w(s_1, s)\},$$
$$\mathcal{A}(V) = \{(s_2, e_2) \in U(\mathbb{C}) \mid s_2 \in S_2 \wedge \forall (s_2, s) \in E, \exists (s, e_1) \in V : e_1 \leq e_2 + w(s_2, s)\},$$

$$\mathsf{Cpre}_{\mathbb{C}}(V) = \mathcal{E}(V) \ \cup \ \mathcal{A}(V). \tag{1}$$

Intuitively, $\mathsf{Cpre}_{\mathbb{C}}(V)$ returns the set of energy levels from which $\mathcal{P}_1$ can force an energy level in $V$ in one step. The operator $\mathsf{Cpre}_{\mathbb{C}}$ is $\subseteq$-monotone over the complete lattice $\mathcal{U}(\mathbb{C})$, and so there exists a *greatest fixed point* for $\mathsf{Cpre}_{\mathbb{C}}$ in the lattice $\mathcal{U}(\mathbb{C})$, denoted by $\mathsf{Cpre}^*_{\mathbb{C}}$. As usual, the greatest fixed point of the operator $\mathsf{Cpre}_{\mathbb{C}}$ can be computed by successive approximations as the last element of the following finite $\subseteq$-descending chain. We define the algorithm $\mathsf{CpreFP}$ that computes this greatest fixed point:

$$U_0 = U(\mathbb{C}),\ U_1 = \mathsf{Cpre}_{\mathbb{C}}(U_0),\ \dots,\ U_n = \mathsf{Cpre}_{\mathbb{C}}(U_{n-1}) = U_{n-1}. \tag{2}$$

The set $U_i$ contains all the energy levels that are sufficient to maintain the energy positive in all dimensions for $i$ steps. Note that the length of this chain can be bounded by $|U(\mathbb{C})|$ and the time needed to compute each element of the chain can be bounded by a polynomial in $|U(\mathbb{C})|$. As a consequence, we obtain the following lemma.

**Lemma 7.** *Let $G = (S_1, S_2, s_{init}, E, k, w)$ be a multi energy game and $\mathbb{C} \in \mathbb{N}$ be a constant. Then $\mathsf{Cpre}^*_{\mathbb{C}}$ can be computed in time bounded by a polynomial in $|U(\mathbb{C})|$, i.e., an exponential in the size of $G$.*

**Symbolic Representation.** To define a symbolic representation of the sets manipulated by the $\mathsf{Cpre}_{\mathbb{C}}$ operator, we exploit the following partial order: let $(s, e), (s', e') \in U(\mathbb{C})$, we define

$$(s, e) \preceq (s', e') \text{ iff } s = s' \text{ and } e \leq e'. \tag{3}$$

A set $V \in \mathcal{U}(\mathbb{C})$ is *closed* if for all $(s, e), (s', e') \in U(\mathbb{C})$, if $(s, e) \in V$ and $(s, e) \preceq (s', e')$, then $(s', e') \in V$. By definition of $\mathsf{Cpre}_{\mathbb{C}}$, we get the following property.

**Lemma 8.** *All sets $U_i$ in eq.* (2) *are closed for $\preceq$.*

Therefore, all sets $U_i$ in the descending chain of eq. (2) can be symbolically represented by their minimal elements $\mathsf{Min}_{\preceq}(U_i)$ which is an antichain of elements for $\preceq$.

Even if the largest antichain can be exponential in $G$, this representation is, in practice, often much more efficient, even for small values of the parameters. For example, with $\mathbb{C} = 4$ and $k = 4$, we have that the cardinality of a set can be as large as $|U_i| \leq 625$ whereas the size of the largest antichain is bounded by $|\mathsf{Min}_{\preceq}(U_i)| \leq 35$. Antichains have proved to be very effective: see for example [1,20,21]. Therefore, our algorithm is expected to have good performance in practice.

**Correctness and Completeness.** The following two lemmas relate the greatest fixed point $\mathsf{Cpre}^*_{\mathbb{C}}$ and the existence of winning strategies for $\mathcal{P}_1$ in $G$.

**Lemma 9 (Correctness).** *Let $G = (S_1, S_2, s_{init}, E, k, w)$ be a multi energy game, let $\mathbb{C} \in \mathbb{N}$ be a constant. If there exists $(c_1, \ldots, c_k) \in \mathbb{N}^k$ s.t. $(s_{init}, (c_1, \ldots, c_k)) \in \mathsf{Cpre}^*_{\mathbb{C}}$, then $\mathcal{P}_1$ has a winning strategy in $G$ for initial credit $(c_1, \ldots, c_k)$ and the memory needed by $\mathcal{P}_1$ can be bounded by $|\mathsf{Min}_{\preceq}(\mathsf{Cpre}^*_{\mathbb{C}})|$ (the size of the antichain of minimal elements in the fixed point).*

Given the set of winning initial credits output by algorithm CpreFP, it is straightforward to derive a corresponding winning strategy of at most exponential size. Indeed, for winning initial credit $\overline{c} \in \mathbb{N}^k$, we build a Moore machine which (i) states are the minimal elements of the fixed point (antichain at most exponential in $G$), (ii) initial state is any element $(t, u)$ among them s.t. $t = s_{init}$ and $u \leq \overline{c}$, (iii) next-action function prescribes an action that ensures remaining in the fixed point, and (iv) update function maintains an accurate energy level in the memory.

**Lemma 10 (Completeness).** *Let $G = (S_1, S_2, s_{init}, E, k, w)$ be a multi energy game in which all absolute values of weights are bounded by $W$. If $\mathcal{P}_1$ has a winning strategy in G and $T = (Q, R)$ is a self-covering tree for G of depth l, then $(s_{init}, (\mathbb{C}, \ldots, \mathbb{C})) \in \mathsf{Cpre}^*_{\mathbb{C}}$ for $\mathbb{C} = 2 \cdot l \cdot W$.*

*Remark 1.* This algorithm is complete in the sense that if a winning strategy exists for $\mathcal{P}_1$, it outputs at least a winning initial credit (and the derived strategy) for $\mathbb{C} = 2 \cdot l \cdot W$. However, this is different from the *fixed initial credit problem*, which consists in deciding if a particular given credit vector is winning and is known to be EXPSPACE-hard [9,26]. In general, there may exist winning credits incomparable to those captured by algorithm CpreFP.

**Incrementality.** While the threshold $2 \cdot l \cdot W$ is sufficient, it may be the case that $\mathcal{P}_1$ can win the game even if its energy level is bounded above by some smaller value. So, in practice, we can use Lemma 9, to justify an incremental algorithm that first starts with small values for the parameter $\mathbb{C}$ and stops as soon as a winning strategy is found or when the value of $\mathbb{C}$ reaches the threshold $2 \cdot l \cdot W$ and no winning strategy has been found.

**Application of the Symbolic Algorithm to MEPGs and MMPGs.** Using the reduction of Lemma 4 that allows us to remove the parity condition, and the equivalence between multi energy games and multi mean-payoff games for finite-memory strategies (given by [15, Theorem 3]), along with Lemma 7 (complexity), Lemma 9 (correctness) and Lemma 10 (completeness), we obtain the following result.

**Theorem 2 (Symbolic and incremental synthesis algorithm).** *Let $G_p$ be a multi energy (resp. multi mean-payoff) parity game. Algorithm* CpreFP *is a symbolic and incremental algorithm that synthesizes a winning strategy in $G_p$ of at most exponential size memory, if a winning (resp. finite-memory winning) strategy exists. In the worst-case, the algorithm* CpreFP *takes exponential time.*

## 5 Trading Finite Memory for Randomness

In this section, we answer the fundamental question regarding the trade-off of memory for randomness in strategies: we study on which kind of games $\mathcal{P}_1$ can replace a pure finite-memory winning strategy by an equally powerful, yet conceptually simpler, randomized memoryless one and discuss how memory is encoded into probability distributions. We summarize our results in Theorem 3 and give a sketch of how they are obtained in the following.

**Energy Games.** Randomization is not helpful for energy objectives, even in one-player games. The proof argument is obtained from the intuition that energy objectives are similar in spirit to safety objectives. Indeed, consider a game fitted with an energy objective, and an almost-sure winning strategy $\lambda_1$. If there exists a single consistent path that violates the energy objective, then there exists a finite prefix witness to violate the energy objective. As the finite prefix has positive probability, and the strategy $\lambda_1$ is almost-sure winning, it follows that no such path exists. In other words, $\lambda_1$ is a sure winning strategy. Since randomization does not help for sure winning strategy, it follows that randomization is not helpful for one-player and two-player energy, multi energy, energy parity and multi energy parity games.

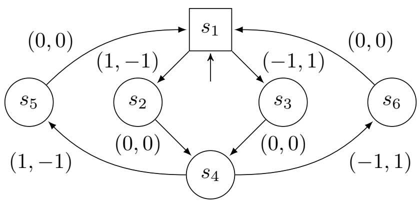


**Fig. 3.** Memory is needed to enforce perfect long-term balance

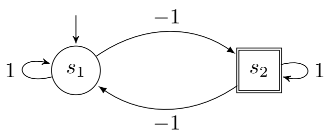


**Fig. 4.** Mixing strategies that are resp. *good for Büchi* and *good for energy*.

**Multi Mean-Payoff (parity) Games.** Randomized memoryless strategies can replace pure finite-memory ones in the one-player multi mean-payoff parity case, but not in the two-player one, even without parity. The fundamental difference between energy and mean-payoff is that energy requires a property to be satisfied *at all times* (in that sense, it is similar to safety), while mean-payoff is a *limit* property. As a consequence, what matters here is the long-run frequencies of weights, not their order of appearance, as opposed to the energy case.

For the one-player case, we extract the frequencies of visit for edges of the graph from the regular outcome that arises from the finite-memory strategy of $\mathcal{P}_1$. We build a randomized strategy with probability distributions on edges that yield the exact same frequencies in the long-run. Therefore, if the original pure finite-memory of $\mathcal{P}_1$ is surely winning, the randomized one is almost-surely winning. For the two-player case, this approach cannot be used as frequencies are not well defined, since the strategy of $\mathcal{P}_2$ is unknown. Consider a game which needs perfect balance between frequencies of appearance of two sets of edges in a play to be winning (Fig. 3). To almost-surely achieve mean-payoff vector $(0, 0)$, $\mathcal{P}_1$ must ensure that the long-term balance between edges $(s_4, s_5)$ and $(s_4, s_6)$ is the same as the one between edges $(s_1, s_3)$ and $(s_1, s_2)$. This is achievable with memory as it suffices to react immediately to compensate the choice of $\mathcal{P}_2$. However, given a randomized memoryless strategy of $\mathcal{P}_1$, $\mathcal{P}_2$ always has a strategy to enforce that the long-term frequency is unbalanced, and thus the game cannot be won almost-surely by $\mathcal{P}_1$ with such a strategy.

**Single Mean-Payoff Parity Games.** Randomized memoryless strategies can replace pure finite-memory ones for single mean-payoff parity games. We prove it in two steps. First, we show that it is the case for the simpler case of *MP Büchi games*. Suppose $\mathcal{P}_1$ has a pure finite-memory winning strategy for such a game. We use the existence of particular pure memoryless strategies on winning states: the classical attractor for Büchi states, and a strategy that ensures that cycles of the outcome have positive energy (whose existence follows from [13]). We build an almost-surely randomized memoryless winning strategy for $\mathcal{P}_1$ by mixing those strategies in the probability distributions, with sufficient probability over the strategy that is good for energy. We illustrate this construction on the simple game $G_p$ depicted on Fig. 4. Let $\lambda_1^{pf} \in \Lambda_1^{PF}$ be a strategy of $\mathcal{P}_1$ s.t. $\mathcal{P}_1$ plays $(s_1, s_1)$ for 8 times, then plays $(s_1, s_2)$ once, and so on. This strategy ensures surely winning for the objective $\phi = \mathsf{MeanPayoff}_{G_p}(3/5)$. Obviously, $\mathcal{P}_1$ has a pure memoryless strategy that ensures winning for the Büchi objective: playing $(s_1, s_2)$. On the other hand, he also has a pure memoryless strategy that ensures cycles of positive energy: playing $(s_1, s_1)$. Let $\lambda_1^{rm} \in \Lambda_1^{RM}$ be the strategy defined as follows: play $(s_1, s_2)$ with probability $\gamma$ and $(s_1, s_1)$ with the remaining probability. This strategy is almost-surely winning for $\phi$ for sufficiently small values of $\gamma$ (e.g., $\gamma = 1/9$).

Second, we extend this result to *MP parity games* using an induction on the number of priorities and the size of games. We consider *subgames* that reduce to the MP Büchi and MP coBüchi (where pure memoryless strategies are known to suffice [16]) cases.

**Summary.** We sum up results for these different classes of games in Theorem 3.

**Theorem 3 (Trading finite memory for randomness).** *The following assertions hold: (1) Randomized strategies are exactly as powerful as pure strategies for energy objectives. Randomized memoryless strategies are not as powerful as pure finite-memory strategies for almost-sure winning in one-player and two-player energy, multi energy, energy parity and multi energy parity games. (2) Randomized memoryless strategies are not as powerful as pure finite-memory strategies for almost-sure winning in two-player multi mean-payoff games. (3) In one-player multi mean-payoff parity games, and two-player single mean-payoff parity games, if there exists a pure finite-memory sure winning strategy, then there exists a randomized memoryless almost-sure winning strategy.*

## 6 Conclusion

In this work, we considered the finite-memory strategy synthesis problem for games with multiple quantitative (energy and mean-payoff) objectives along with a parity objective. We established tight (matching upper and lower) exponential bounds on the memory requirements for such strategies (Theorem 1), significantly improving the previous triple exponential bound for multi energy games (without parity) that could be derived from results in literature for games on VASS. We presented an optimal symbolic and incremental strategy synthesis algorithm (Theorem 2). Finally, we also presented a precise characterization of the trade-off of memory for randomness in strategies (Theorem 3).

## References

1. Abdulla, P.A., Chen, Y.-F., Holík, L., Mayr, R., Vojnar, T.: When Simulation Meets Antichains. In: Esparza, J., Majumdar, R. (eds.) TACAS 2010. LNCS, vol. 6015, pp. 158–174. Springer, Heidelberg (2010)
2. Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. J. ACM 49(5), 672–713 (2002)
3. Bernet, J., Janin, D., Walukiewicz, I.: Permissive strategies: from parity games to safety games. ITA 36(3), 261–275 (2002)
4. Bloem, R., Chatterjee, K., Henzinger, T.A., Jobstmann, B.: Better Quality in Synthesis through Quantitative Objectives. In: Bouajjani, A., Maler, O. (eds.) CAV 2009. LNCS, vol. 5643, pp. 140–156. Springer, Heidelberg (2009)
5. Bloem, R., Greimel, K., Henzinger, T.A., Jobstmann, B.: Synthesizing robust systems. In: Proc. of FMCAD, pp. 85–92. IEEE (2009)
6. Borosh, I., Treybig, B.: Bounds on positive integral solutions of linear diophantine equations. Proc. of the American Mathematical Society 55(2), 299–304 (1976)
7. Bouyer, P., Fahrenberg, U., Larsen, K.G., Markey, N.: Timed automata with observers under energy constraints. In: Proc. of HSCC, pp. 61–70. ACM (2010)
8. Bouyer, P., Markey, N., Olschewski, J., Ummels, M.: Measuring Permissiveness in Parity Games: Mean-Payoff Parity Games Revisited. In: Bultan, T., Hsiung, P.-A. (eds.) ATVA 2011. LNCS, vol. 6996, pp. 135–149. Springer, Heidelberg (2011)
9. Brázdil, T., Jančar, P., Kučera, A.: Reachability Games on Extended Vector Addition Systems with States. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) ICALP 2010, Part II. LNCS, vol. 6199, pp. 478–489. Springer, Heidelberg (2010)
10. Černý, P., Chatterjee, K., Henzinger, T.A., Radhakrishna, A., Singh, R.: Quantitative Synthesis for Concurrent Programs. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 243–259. Springer, Heidelberg (2011)
11. Cerný, P., Henzinger, T.A., Radhakrishna, A.: Simulation distances. Theor. Comput. Sci. 413(1), 21–35 (2012)
12. Chakrabarti, A., de Alfaro, L., Henzinger, T.A., Stoelinga, M.: Resource Interfaces. In: Alur, R., Lee, I. (eds.) EMSOFT 2003. LNCS, vol. 2855, pp. 117–133. Springer, Heidelberg (2003)
13. Chatterjee, K., Doyen, L.: Energy Parity Games. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) ICALP 2010, Part II. LNCS, vol. 6199, pp. 599–610. Springer, Heidelberg (2010)
14. Chatterjee, K., Doyen, L., Henzinger, T.A.: Quantitative languages. ACM Trans. Comput. Log. 11(4) (2010)
15. Chatterjee, K., Doyen, L., Henzinger, T.A., Raskin, J.-F.: Generalized mean-payoff and energy games. In: Proc. of FSTTCS. LIPIcs, vol. 8, pp. 505–516. Schloss Dagstuhl - LZI (2010)
16. Chatterjee, K., Henzinger, T.A., Jurdzinski, M.: Mean-payoff parity games. In: Proc. of LICS, pp. 178–187. IEEE Computer Society (2005)
17. Chatterjee, K., Randour, M., Raskin, J.-F.: Strategy synthesis for multi-dimensional quantitative objectives. CoRR, abs/1201.5073 (2012), `http://arxiv.org/abs/1201.5073`

18. Church, A.: Logic, arithmetic, and automata. In: Proceedings of the International Congress of Mathematicians, pp. 23–35. Institut Mittag-Leffler (1962)
19. de Alfaro, L., Henzinger, T.A.: Interface Theories for Component-Based Design. In: Henzinger, T.A., Kirsch, C.M. (eds.) EMSOFT 2001. LNCS, vol. 2211, pp. 148–165. Springer, Heidelberg (2001)
20. De Wulf, M., Doyen, L., Henzinger, T.A., Raskin, J.-F.: Antichains: A New Algorithm for Checking Universality of Finite Automata. In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 17–30. Springer, Heidelberg (2006)
21. Doyen, L., Raskin, J.-F.: Antichain Algorithms for Finite Automata. In: Esparza, J., Majumdar, R. (eds.) TACAS 2010. LNCS, vol. 6015, pp. 2–22. Springer, Heidelberg (2010)
22. Doyen, L., Raskin, J.-F.: Games with imperfect information: Theory and algorithms. In: Lectures in Game Theory for Computer Scientists, pp. 185–212 (2011)
23. Ehrenfeucht, A., Mycielski, J.: Positional strategies for mean payoff games. Int. Journal of Game Theory 8(2), 109–113 (1979)
24. Emerson, E.A., Jutla, C.: The complexity of tree automata and logics of programs. In: Proc. of FOCS, pp. 328–337. IEEE (1988)
25. Emerson, E.A., Jutla, C.: Tree automata, mu-calculus and determinacy. In: Proc. of FOCS, pp. 368–377. IEEE (1991)
26. Fahrenberg, U., Juhl, L., Larsen, K.G., Srba, J.: Energy Games in Multiweighted Automata. In: Cerone, A., Pihlajasaari, P. (eds.) ICTAC 2011. LNCS, vol. 6916, pp. 95–115. Springer, Heidelberg (2011)
27. Grädel, E., Thomas, W., Wilke, T. (eds.): Automata, Logics, and Infinite Games. LNCS, vol. 2500. Springer, Heidelberg (2002)
28. Gurevich, Y., Harrington, L.: Trees, automata, and games. In: Proc. of STOC, pp. 60–65. ACM (1982)
29. Henzinger, T.A., Kupferman, O., Rajamani, S.: Fair simulation. Information and Computation 173(1), 64–81 (2002)
30. Martin, D.A.: The determinacy of Blackwell games. The Journal of Symbolic Logic 63(4), 1565–1581 (1998)
31. Pnueli, A., Rosner, R.: On the synthesis of a reactive module. In: Proc. of POPL, pp. 179–190 (1989)
32. Rackoff, C.: The covering and boundedness problems for vector addition systems. Theor. Comput. Sci. 6, 223–231 (1978)
33. Ramadge, P.J., Wonham, W.M.: Supervisory control of a class of discrete-event processes. SIAM Journal of Control and Optimization 25(1), 206–230 (1987)
34. Rosier, L.E., Yen, H.-C.: A multiparameter analysis of the boundedness problem for vector addition systems. J. Comput. Syst. Sci. 32(1), 105–135 (1986)
35. Thomas, W.: Languages, automata, and logic. In: Handbook of Formal Languages, vol.3: Beyond Words, ch. 7, pp. 389–455. Springer (1997)
36. Vardi, M.Y.: Automatic verification of probabilistic concurrent finite-state programs. In: Proc. of FOCS, pp. 327–338. IEEE Computer Society (1985)
37. Velner, Y., Rabinovich, A.: Church Synthesis Problem for Noisy Input. In: Hofmann, M. (ed.) FOSSACS 2011. LNCS, vol. 6604, pp. 275–289. Springer, Heidelberg (2011)
38. Zielonka, W.: Infinite games on finitely coloured graphs with applications to automata on infinite trees. Theoretical Computer Science 200(1-2), 135–183 (1998)
39. Zwick, U., Paterson, M.: The complexity of mean payoff games on graphs. Theoretical Computer Science 158, 343–359 (1996)