Pareto Curves of Multidimensional Mean-Payoff Games

Romain Brenguier[(✉)] and Jean-François Raskin

Université Libre de Bruxelles (U.L.B.), Brussel, Belgium
romain.brenguier@ulb.ac.be

Abstract. In this paper, we study the set of thresholds that the protagonist can force in a zero-sum two-player multidimensional mean-payoff game. The set of maximal elements of such a set is called the *Pareto curve*, a classical tool to analyze *trade-offs*. As thresholds are vectors of real numbers in multiple dimensions, there exist usually an infinite number of such maximal elements. Our main results are as follow. First, we study the geometry of this set and show that it is definable as a finite union of convex sets given by linear inequations. Second, we provide a $\Sigma_2 \mathsf{P}$ algorithm to decide if this set intersects a convex set defined by linear inequations, and we prove the optimality of our algorithm by providing a matching complexity lower bound for the problem. Furthermore, we show that, under natural assumptions, i.e. fixed number of dimensions and polynomially bounded weights in the game, the problem can be solved in deterministic polynomial time. Finally, we show that the Pareto curve can be effectively constructed, and under the former natural assumptions, this construction can be done in deterministic polynomial time.

1 Introduction

Two-player zero-sum games played on graphs are adequate models for open reactive systems [12], i.e. systems maintaining a continuous interaction with their environment. In such model, Eve (the protagonist) models the system, Adam (the antagonist) models the environment, and a winning strategy for Eve in this game represents a controller that enforces a good property (modeled as the winning condition in the game) against all possible behaviors of the environment. Recently, there has been a large effort to study quantitative extensions of those graph games, see e.g. [6]. Those extensions are useful to model quantitative aspects of reactive systems such as mean energy or peak energy consumption, mean response time, etc. In practice, a system is most often exhibiting several such quantitative aspects, and they may be conflicting, e.g. one may need to consume more energy in order to ensure of a lower mean response time. This is why there is a clear need to study multi-dimensional quantitative games.

In [15], the *threshold problem* for multi-dimensional mean-payoff games is studied, i.e. given a d-dimensional value vector $v \in \mathbb{R}^d$, does Eve have a strategy

against all strategies of Adam to enforce values larger or equal to v. As weights in the game are given as vectors in multiple dimensions, there are usually an infinite number of incomparable thresholds that Eve is able to enforce. The set of maximal thresholds that Eve can enforce is called the *Pareto curve*, it is the classical tool to analyze *trade-offs*. Another application of the Pareto curve is the study of multiplayer games. For instance to compute Nash equilibria, a multiplayer game with mean-payoff objectives is transformed into a multidimensional mean-payoff two-player game [2], and the Pareto curve of this multidimensional game allows us to compute the equilibria of the original multiplayer game. In this paper, we study the Pareto curve and the set of thresholds that Eve can enforce in a multidimensional mean-payoff games.

Contributions. To effectively analyze the trade-offs in systems formalized by multidimensional mean-payoff games, we need algorithms to answer queries about Pareto curves or to compute an effective representation of them. This is the subject of this paper. Our main contributions are as follows.

First, we characterize the geometry of the set of thresholds that Eve can force: we show that this infinite set can be effectively represented as a (finite) union of convex sets defined by linear inequations. We obtain this result both for games where the mean-payoff is given dimension by dimension using lim inf (Theorem 4), and for a mixture of lim inf and lim sup (Theorem 10). Using this symbolic representation as a finite union of convex sets, it is now possible for instance to optimize linear functions by calls to linear programming.

Second, we study the computational complexity of natural associated decision problems. We provide a $\Sigma_2 P$ algorithm to decide if this set of thresholds intersects a convex set defined by linear inequations, and we prove the optimality of our algorithm by providing a matching complexity lower bound for the problem (Theorem 6). To obtain this result and several others in our paper, we extensively use techniques from discrete geometry [9] but we also need to establish new non-trivial results. In particular, we provide new results on the complexity of manipulating and querying linear sets defined by sets of linear inequations (Theorem 3). We believe that those results are of interest on their own. Equipped with those new results, we show that, even if the Pareto curve is represented by an exponential number of convex sets, each of them being defined by an exponential number of linear inequations, they are well behaved. Indeed, all the inequations that are needed to represent the Pareto curve and its downward closure (the set of thresholds that can be forced by Eve), have encoding that are bounded by polynomial functions in the size of the game.

Third, we show that it is possible to answer queries on the set thresholds that Eve can force (Theorem 7) and to construct the Pareto curve (Theorem 8) in *deterministic polynomial time* for fixed number of dimensions and polynomially bounded weights. Those results are of practical relevance as the number of dimension while multiple is often quite low in practice, and polynomially bounded weight is also a reasonable assumption, see [4,8,16] for papers where those two properties are exploited.

Related Works. In [1], Alur et al. consider languages of infinite words definable by Boolean queries over multidimensional mean-payoff automata. They study the accumulation points of infinite runs as a way to define an acceptance condition. They do not consider the construction of the Pareto curves associated to languages. Here, we show how to construct the Pareto curves in the more general and challenging setting of multidimensional mean-payoff games.

In [11], Papadimitriou et al. define a general procedure to construct *approximations* of Pareto curves. For models with *fixed* number of dimensions, they identify conditions that are sufficient to ensure that this approximation can be constructed in polynomial time. This technique has been used e.g. to provide approximate constructions of the Pareto curves for discounted sum Markov decision processes [7]. With the technique of [11], we can obtain approximations of the Pareto curves of multidimensional mean-payoff games in polynomial time for fixed number of dimensions. Here we provide a stronger result as we show how to construct *exact* representations of the Pareto curves (and not only approximations!) of multidimensional mean-payoff games in deterministic polynomial time for fixed number of dimensions.

Structure of the Paper. Section 2 defines the problems that we solve. Section 3 establishes general complexity results on the geometric objects that we use in the core of our paper. Section 4 solves the lim inf case. We concentrate on this case first as it exhibits all the difficulties of the general case with simpler notations. Section 5 deals with the construction of a concrete representation of the Pareto curve. In Sect. 6 we solve the general problem in which lim inf and lim sup are mixed.

For lack of space, the technical proofs are omitted, and can be found in [3].

2   Preliminaries

Arenas. We define *arenas* for two players that we call Eve and Adam. An *arena* $\mathcal{A}$ is a tuple $\langle \mathsf{States}_\exists, \mathsf{States}_\forall, \mathsf{Edges} \rangle$, where:

- $\mathsf{States} = \mathsf{States}_\exists \uplus \mathsf{States}_\forall$ is a finite set of *states* partitioned between the states of Eve and those of Adam;[1]
- $\mathsf{Edges} \subseteq \mathsf{States} \times \mathsf{States}$ is the set of *edges*. W.l.o.g. we assume that for all $s \in \mathsf{States}$, there exists $s' \in \mathsf{States}$ such that $(s, s') \in \mathsf{Edges}$.

A play proceeds as follows. Whenever we arrive at a state s: if $s \in \mathsf{States}_\exists$, then Eve selects a state s' such that $(s, s') \in \mathsf{Edges}$; if $s \in \mathsf{States}_\forall$, then Adam selects a state s' such that $(s, s') \in \mathsf{Edges}$. The game then continues from s' and this is repeated to form an infinite sequence of states. Formally, a *play* in the arena $\mathcal{A}$ is an infinite sequence of states $\rho = \rho_0 \rho_1 \cdots$ such that for all $i \geq 0$, $(\rho_i, \rho_{i+1}) \in \mathsf{Edges}$. We write $\rho_{\leq n}$ for the prefix $\rho_0 \cdots \rho_n$. A *history* h of the arena $\mathcal{A}$ is a (finite and non-empty) prefix of a play, i.e. an element of $\mathsf{States}^* \cdot \mathsf{States}$.

---

[1] We will write $|\mathsf{States}|$ for the cardinal of States.

Strategies. Let $\mathcal{A}$ be an arena, a *strategy* for Eve maps histories ending in a state of $\mathsf{States}_\exists$ to a successor of that state. Formally, it is a function $\sigma_\exists \colon \mathsf{States}^* \cdot \mathsf{States}_\exists \to \mathsf{States}$, such that for all histories h and states s, $(s, \sigma_\exists(h \cdot s)) \in \mathsf{Edges}$. Similarly, a *strategy* for Adam is a function $\sigma_\forall \colon \mathsf{States}^* \cdot \mathsf{States}_\forall \to \mathsf{Actions}$, such that for all for all histories h and states s, $(s, \sigma_\forall(h \cdot s)) \in \mathsf{Edges}$. A strategy $\sigma_\forall$ is *memoryless* if for all histories h and h', and all states s, $\sigma_\forall(h \cdot s) = \sigma_\forall(h' \cdot s)$. We write $\mathbb{M}$ for the (finite) set of memoryless strategies of Adam. Let $\sigma_\exists$ be a strategy for Eve, a play ρ is *compatible* with the strategy $\sigma_\exists$ if, for all $k \geq 0$, if $\rho_k \in \mathsf{States}_\exists$ then $\rho_{k+1} = \sigma_\exists(\rho \leq k)$. We write $\mathsf{Outcome}_\mathcal{A}(s, \sigma_\exists)$ for the set of plays in $\mathcal{A}$ that are compatible with strategy $\sigma_\exists$ and have initial state s (i.e. ρ such that $\rho_0 = s$). These plays are called *outcomes* of $\sigma_\exists$ from s. We simply write $\mathsf{Outcome}(s, \sigma_\exists)$ when $\mathcal{A}$ is clear from context. The set of outcomes $\mathsf{Outcome}_\mathcal{A}(s, \sigma_\forall)$ of a strategy of Adam is defined symmetrically.

Weighted Game. A *weighted game* $\mathcal{G} = \langle \mathcal{A}, w, I, J \rangle$ is an arena $\mathcal{A}$ equipped with a *weight function* $w \colon \mathsf{Edges} \mapsto \mathbb{Z}^d$, and a partition of the set of dimensions $[\![1, d]\!] = \{1, 2, \dots, d\}$ into $I \uplus J = [\![1, d]\!]$. We call d the dimension of $\mathcal{G}$. Given a weight function w, we write w_i for the projection to the i-th dimension of the function w. We write $W_\mathcal{G}$ for the maximal absolute value appearing in the weights: $W_\mathcal{G} = \max\{|w_i(e)| \mid i \in [\![1, d]\!], e \in \mathsf{Edges}\}$. The *mean-payoff inferior* and *mean-payoff superior* over dimension i of a play ρ are given by:

$$\underline{\mathsf{MP}}_i(\rho) = \liminf_{n \to \infty} \frac{1}{n} \sum_{0 \leq k < n} w_i(\rho_k, \rho_{k+1}),$$

$$\overline{\mathsf{MP}}_i(\rho) = \limsup_{n \to \infty} \frac{1}{n} \sum_{0 \leq k < n} w_i(\rho_k, \rho_{k+1}).$$

The goal of Eve is to maximize the mean-payoff inferior for the dimensions in I, and the mean-payoff superior for the dimensions in J. Let $\mathcal{G}$ be a weighted game, s a state of $\mathcal{G}$, and $v \in \mathbb{R}^d$, we say that a strategy $\sigma_\exists$ *ensures thresholds* v from state s if for all outcomes $\rho \in \mathsf{Outcome}_\mathcal{A}(s, \sigma_\exists)$, for all dimensions $i \in I$, $\underline{\mathsf{MP}}_i(\rho) \geq v_i$, and for all dimensions $j \in J$, $\overline{\mathsf{MP}}_j(\rho) \geq v_j$.

Pareto Optimality. We are interested in strategies of Eve that ensure thresholds as high as possible on all dimensions. However, since the weights are multidimensional, there is not a unique maximal threshold in general. We use the concept of *Pareto optimality* to identify the most interesting thresholds. To define the set of Pareto optimal thresholds, we first define the set of thresholds that Eve can force:

$$\mathtt{value}(\mathcal{G}, s) = \left\{ v \in \mathbb{R}^d \mid \exists \sigma_\exists \cdot \forall \rho \in \mathsf{Outcome}(s, \sigma_\exists) \cdot \begin{array}{l} \forall i \in I : \underline{\mathsf{MP}}_i(\rho) \geq v_i \\ \wedge \forall j \in J : \overline{\mathsf{MP}}_j(\rho) \geq v_j \end{array} \right\}.$$

A threshold $v \in \mathbb{R}^d$ is *Pareto optimal* from s if is maximal in the set $\mathtt{value}(\mathcal{G}, s)$. So the set of Pareto optimal thresholds is defined as:

$$\mathsf{PO}(\mathcal{G}, s) = \{ v \in \mathtt{value}(\mathcal{G}, s) \mid \neg \exists v' \in \mathtt{value}(\mathcal{G}, s) : v' > v \}.$$

We refer to this set as the *Pareto curve* of the game. Our goal is to compute a representation of this curve. Note that the set of thresholds that Eve can force is exactly equal to the downward closure of the Pareto optimal thresholds, i.e. $\mathtt{value}(\mathcal{G}, s) = \downarrow \mathsf{PO}(\mathcal{G}, s)$.

Linear Inequations. Let $a \in \mathbb{Q}^d$ be a vector in d dimensions. The associated *linear function* $\alpha_a \colon \mathbb{R}^d \mapsto \mathbb{R}$ is the function $\alpha_a(x) = \sum_{i \in [\![1,d]\!]} a_i \cdot x_i$ that computes the weighted sum relative to a. A *linear inequation* is a pair (a, b) where $a \in \mathbb{Q}^d \setminus \{\mathbf{0}\}$ and $b \in \mathbb{Q}$. The *half-space* satisfying (a, b) is the set $\frac{1}{2}\mathtt{space}(a, b) = \{x \in \mathbb{R}^d \mid \alpha_a(x) \geq b\}$. A *linear equation* is also given by a pair (a, b) where $a \in \mathbb{Q}^d \setminus \{\mathbf{0}\}$ and $b \in \mathbb{Q}$ but we associate to it the *hyperplane* $\mathtt{hplane}(a, b) = \{x \in \mathbb{R}^d \mid \alpha_a(x) = b\}$. If $H = \frac{1}{2}\mathtt{space}(a, b)$ is a half-space, we sometimes write $\mathtt{hplane}(H)$ for the *associated hyperplane* $\mathtt{hplane}(a, b)$. A *system of linear inequations* is a set $\lambda = \{(a_1, b_1), \ldots, (a_l, b_l)\}$ of linear inequations. The *polyhedron* generated by λ is the set $\mathtt{polyhedron}(\lambda) = \bigcap_{(a,b) \in \lambda} \frac{1}{2}\mathtt{space}(a, b)$.

A natural problem, is to try to optimize the threshold we can ensure with respect to a linear function $\alpha \colon \mathbb{R}^d \mapsto \mathbb{R}$. We are looking for a strategy $\sigma_\exists$ which ensures a threshold $v \in \mathbb{R}^d$, and such that there is no $\sigma'_\exists$ which ensures a threshold $v' \in \mathbb{R}^d$, with $\alpha(v') > \alpha(v)$. To make this into a decision problem, we fix a real b, and ask if it is possible to ensure threshold v such that $\alpha(v) \geq b$. We consider a generalization of this problem which considers a set of linear inequations instead of a single one.

Polyhedron Value Problem. Given a mean-payoff game $\mathcal{G}$, a set of linear inequations λ over elements of $\mathbb{R}^d$, the *polyhedron value problem* asks whether there is a strategy $\sigma_\exists$ and a value $v \in \mathtt{polyhedron}(\lambda)$ such that $\sigma_\exists$ ensures v. Note that this is equivalent to ask whether $\mathtt{polyhedron}(\lambda)$ intersects $\mathtt{value}(\mathcal{G}, s)$.

Remark 1. Other works ([8,15] for instance) focus on the 0-value problem, which is a special case of the polyhedron value problem (take as polyhedron the set $\mathbb{R}^d_+$). This special case is simpler: we will show that the polyhedron value problem is $\Sigma_2\mathsf{P}$-complete while the 0-value problem is coNP-complete [15].

Consider a system with n resources $R_1, \ldots, R_n$ that are shared among d agents $A_1, \ldots, A_d$. Two agents cannot access the same resource at the same time and can request one resource at any time. We want to control the access to the resources in a way that minimizes the time that is spent during the waiting period by the different agents. This situation can be seen as a d dimensional game, in which if A_i is waiting then the reward is -1 on the i-th dimension and 0 otherwise. A situation with two agents and one resource is represented in Fig. 1.

On each dimension, the average corresponds to the opposite of the average waiting time of each agent. For limit inferior objectives the controller cannot ensure a payoff of 0 on all dimensions. However, it can ensure thresholds like $(-1, 0)$, $(0, -1)$, or $(-\frac{1}{2}, -\frac{1}{2})$, and in fact all the thresholds on the line segment from point $(-1, 0)$ to $(0, -1)$, or below it (this set is the set of *feasible thresholds*).
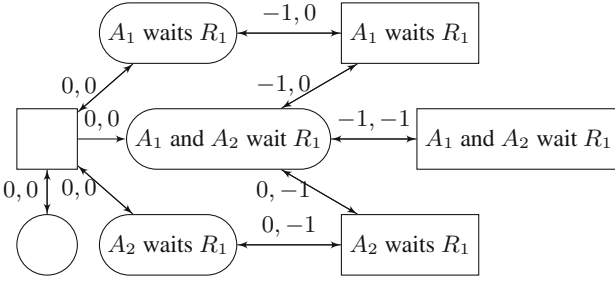
Fig. 1. A two-dimensional mean-payoff game. Rounded states belong to `Eve` and rectangles to `Adam`.
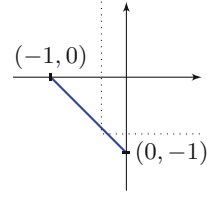
Fig. 2. Pareto curve of the game of Fig. 1.

Figure 2 shows the Pareto curve of the game. To illustrate the polyhedron value problem, assume we want a strategy which gives at least $-\frac{1}{3}$ on the first dimension, at least $-\frac{3}{4}$ on the second one: this corresponds to solution of the problems with $\lambda = ((1,0), -\frac{1}{3}), ((0,1), -\frac{3}{4})$. The frontier of this polyhedron is represented by dotted lines on the figure. This polyhedron has a non-empty intersection with the set of feasible thresholds, which means the problem has a solution.

3   Geometrical Representations

Since our typical reader may not be familiar with all the notions of discrete geometry that we need, we summarize in this section useful notions and properties related to convex sets which are useful for our characterization of the sets $\mathsf{PO}(\mathcal{G}, s)$ and $\mathtt{value}(\mathcal{G}, s)$. For an introduction to discrete geometry, we refer the interested reader to [9]. We also prove new results in Theorems 1, 2 and 3 on manipulating and querying polytopes and systems of linear inequations. Those results are necessary to prove the main theorems of our paper, and we believe that they are of interest on their own.

To allow computational complexity measure, the size of the representations of geometrical objects is relevant. We give here the number of bits required to represent the objects that we manipulate. The *size* of a rational number $r = \frac{p}{q} \in \mathbb{Q}$ where $p \in \mathbb{Z}, q \in \mathbb{N}$, p and q are relatively prime, is: $||r|| = 1 + \lceil \log_2(|p|+1) \rceil + \lceil \log_2(q + 1) \rceil$. The size of a vector $v = (r_1, \ldots, r_d)$ is $||v|| = d + \sum_{i \in [\![1,d]\!]} ||r_i||$. The size of an equation (a, b) is $||(a,b)|| = ||a|| + ||b||$ and the size of a system of equations λ is $||\lambda|| = \sum_{(a,b) \in \lambda} ||(a,b)||$.

A bounded polyhedron is called a *polytope*. A *face* F of P is a subset of P of the form $F = P \cap H_F$, where H_F is a half-space such that $P \subseteq H_F$. In that case, say that H_F *defines* face F of P. A face of dimension 1 is called a *vertex*. If P has dimension d', then a face of dimension $d' - 1$ is called a *facet*. Given a polytope P, a *complete set of facet-defining half-spaces* $\mathcal{F}$ contains for each facet F a half-space $H_F = \frac{1}{2}\mathtt{space}(a_F, b_F)$ such that $P \cap \mathtt{hplane}(a_F, b_F) = F$ and $P \subseteq H_F$.

We will write $\mathcal{F}(P)$ for such a set. The *convex hull* of a set of points $X \subseteq \mathbb{R}^d$ is the set $\mathtt{conv}(X) = \left\{ \sum_{x \in X} t_x \cdot x \mid \forall x \in X.\ t_x \in [0,1] \wedge \sum_{x \in X} t_x = 1 \right\}$. The *downward closure* of a set of points $X \subseteq \mathbb{R}^d$, is the set $\downarrow X = \{ x \mid \exists x' \in X.\ \forall i \in [\![1,d]\!].\ x_i \leq x_i' \}$.

If X is a finite set of points, the convex hull $P = \mathtt{conv}(X)$ can be written as a finite intersection of half-spaces [9], it is therefore a polytope. It can be represented either by its extremal points or as the intersection of its facet-defining half-spaces.

For our algorithms, it is important to be able to go from the half-space representation to the extremal point representation and vice-versa. We need also to bound the complexity of the objects that we obtain, i.e. we want to ensure that the half-spaces are defined with inequations of polynomial size and the extremal points to be representable with polynomial encodings. We will show in Theorem 3 that this is possible.

Small Solutions of Large Systems of Equations. The following theorem establishes that if a system of linear equations has a solution, then it also has a solution with a small encoding.

Theorem 1. *There is a polynomial function P_1 such that for all system of equations λ of $\mathbb{R}^d$, if $\bigcap_{(a,b) \in \lambda} \mathtt{hplane}(a,b) \neq \varnothing$, then there exists $x \in \bigcap_{(a,b) \in \lambda} \mathtt{hplane}(a,b)$ whose representation has size smaller than $P_1(d) \cdot (1 + \max\{||(a,b)|| \mid (a,b) \in \lambda\})$.*

The proof given in long version of this paper relies on a result of [10] that we extend to non-singular, and non-square matrices of rational numbers rather than integers. Note also that our bound depends on the number of dimension d but not on the number of equations as in [10].

Small Witnesses of Large Systems of Inequations. To decide the polyhedron value problem, our algorithm nondeterministically constructs solutions of large systems of inequations. We show in Theorem 2 that we can restrict the guesses to points whose representation is of polynomial size. The proof relies on Lemma 1 that says that if a system of inequations has a solution, then there is one at the intersections of at most d of the hyperplanes defined by the associated equations. This is illustrated in Fig. 3: in two dimensions, if a collection of half-spaces (i.e. half-planes here) intersect (green shaded area in the picture), then either there is a point at the intersection of two boundary lines which is in the intersection (this is the case

Fig. 3. Illustration of Lemma 1 in the case of two dimensions: In this example, the possible witnesses of the property are circled.

in our example for the blue points), or one of these lines is included in the intersection (this would be the case for instance if we only took parallel lines).
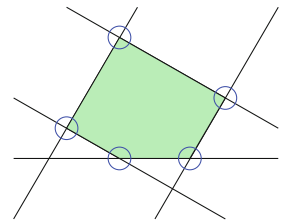
Lemma 1. *Let $H_1, \ldots, H_n$ be n inequations of $\mathbb{R}^d$. If $\bigcap_{i=1}^{n} \frac{1}{2} \mathtt{space}(H_i) \neq \varnothing$ then there are $k \leq d$ indexes $i_1, \ldots, i_d$ such that:*

1. $\bigcap_{j=1}^{k} \mathtt{hplane}(H_{i_j}) \neq \varnothing$, *and*
2. $\bigcap_{j=1}^{k} \mathtt{hplane}(H_{i_j}) \subseteq \bigcap_{i=1}^{n} \frac{1}{2} \mathtt{space}(H_i)$.

From Lemma 1, we conclude that small solutions always exists for systems of linear inequations independently of the number of inequations. Note that the main difference between the next theorem and Theorem 1 is that we consider here systems of inequations rather than equations.

Theorem 2. *There is a polynomial function P_1 such that for all systems of inequations λ of $\mathbb{R}^d$, if $\mathtt{polyhedron}(\lambda) \neq \varnothing$ then there is a point $x \in \mathtt{polyhedron}(\lambda)$ whose representation has size smaller than $P_1(d) \cdot (1 + \max\{||(a,b)|| \mid (a,b) \in \lambda\})$.*

Size Obtained When Changing the Representation of Polyhedra. As already recalled, it is well known that we can represent a polytope either as the intersection of half-spaces (solutions of a system of inequations) or by the finite set of its vertices (extremal points). Theorem 3 characterizes the complexity of one representation w.r.t. the other. Point 1 tells us how to bound the size of the inequations in the half-spaces representation as a function of the size of the representation of the points in the vertices representation. Point 2 does the same for the downward closure of the convex hull of the set of points. Point 3 tells us how to bound the size of the representation of the vertices as a function of the size of the inequations in the half-space representation. Proofs can be found in the long version of this paper.

Theorem 3. *There are polynomial functions P_2 and P_3 such that:*

1. *given a finite set of points $V = \{v_1, \ldots, v_n\}$, there are $k \leq n^d$ inequations $(a_1, b_1), \ldots, (a_k, b_k)$ whose representations have size smaller than $P_2(d) \cdot (2 + \log_2(\max\{||v|| \mid v \in V\}))$ and such that $\bigcap_{i \in [\![1,k]\!]} \frac{1}{2} \mathtt{space}(a_i, b_i) = \mathtt{conv}(V)$.*
2. *given a finite set of points $V = \{v_1, \ldots, v_n\}$, there are $k \in \mathbb{N}$ inequations $(a_1, b_1), \ldots, (a_k, b_k)$ whose representations have size smaller than $P_2(d) \cdot (2 + \log_2(\max\{||v|| \mid v \in V\}))$ and such that $\bigcap_{1 \leq i \leq k} \frac{1}{2} \mathtt{space}(a_i, b_i) = \downarrow \mathtt{conv}(V)$.*
3. *given a polytope P (i.e. a bounded polyhedron) represented by a system of inequations λ, such that $P = \bigcap_{(a,b) \in \lambda} \frac{1}{2} \mathtt{space}(a, b)$, there is a finite set V of points whose representations have size smaller than $P_3(d) \cdot (2 + \log_2(\max\{||(a,b)|| \mid (a,b) \in \lambda\}))$ and such that $\mathtt{conv}(V) = P$.*

4 The Limit Inferior Case

Let us fix for this section a weighted game $\mathcal{G} = \langle \mathsf{States}_{\exists}, \mathsf{States}_{\forall}, \mathsf{Edges}, w, I, J \rangle$ with $J = \emptyset$, i.e. the averages for all dimensions are defined using lim inf. In this case, the set of thresholds that can be ensured by Eve from state $s \in \mathsf{States}$ is:

$$\mathtt{value}(\mathcal{G}, s) = \left\{ v \in \mathbb{R}^d \mid \exists \sigma_{\exists}. \; \forall \rho \in \mathsf{Outcome}(s, \sigma_{\exists}). \; \forall i \in [\![1, d]\!]. \; \underline{\mathsf{MP}}_i(\rho) \geq v_i \right\}.$$

To obtain a geometrical characterization of this set, we first study the set of thresholds that Eve can ensure when Adam plays according to a fixed memoryless strategy. Then, we show that the set $\mathtt{value}(\mathcal{G}, s)$ is the intersection of those sets for all the memoryless strategies of Adam. With the results of previous section, we deduce that if there is a solution to the *polyhedron value problem*, then there is one of bounded size which allows us to justify the correctness of a $\Sigma_2\mathsf{P}$-algorithm for this problem. Finally, we show that this algorithm has optimal worst-case complexity by providing a matching $\Sigma_2\mathsf{P}$-lower bound.

Playing Against Memoryless Strategies of Adam. Memoryless strategies for Adam are important as they are optimal for the threshold problem [15], i.e. if Adam has a strategy to prevent Eve from ensuring some threshold v then he has a memoryless one to do so. Our analysis relies on simple cycles. Let $S \subseteq \mathsf{States}$ be a subset of states of the arena of $\mathcal{G}$. A *simple cycle* within S is a finite sequence of states $s_0 \cdot s_1 \cdots s_n \in S^*$, such that $s_0 = s_n$, and for all i and j, $0 \leq i < j < n$, $s_i \neq s_j$. We write $\mathbb{C}(S)$ for the set of simple cycles of $\mathcal{A}$ within S. Let $\sigma_\forall \in \mathbb{M}$ be a memoryless strategy for Adam, this strategy induces the graph $\mathcal{G}(\sigma_\forall) = \langle \mathsf{States}, \mathsf{Edges}_{\sigma_\forall} \rangle$ where $\mathsf{Edges}_{\sigma_\forall} = \{(s, s') \in \mathsf{Edges} \mid s \in \mathsf{States}_\exists \vee (s \in \mathsf{States}_\forall \wedge \sigma_\forall(s) = s')\}$ which is a subgraph of the game arena in which Adam plays according to the memoryless strategy $\sigma_\forall$. We denote by $\mathtt{SCC}(s, \sigma_\forall)$ the set of *strongly connected components* accessible from s in $\mathcal{G}(\sigma_\forall)$.

Lemma 2. *For all $\sigma_\forall \in \mathbb{M}$, for all infinite paths $\rho = \rho_0 \rho_1 \ldots \rho_n \ldots$ in $\mathcal{G}(\sigma_\forall)$, let $S \in SCC(\rho_0, \sigma_\forall)$ be such that $\mathsf{Inf}(\rho) \subseteq S$, then*

$$\underline{\mathsf{MP}}(\rho) \in \downarrow \mathit{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\} \right).$$

Proof. An *accumulation point* of a sequence $x_0, x_1, \ldots, x_n, \ldots$ of vectors in $\mathbb{R}^d$ is a vector $x \in \mathbb{R}^d$ such that for every open set containing x, there are infinitely many elements in the sequence which belong to the open set. It is proved in [1] that if a run ρ gets trapped for ever in the SCC S (i.e. $\mathsf{Inf}(\rho) \subseteq S$) the set of accumulation points of the sequence $(\frac{1}{n} \cdot w(\rho \leq n))_{\leq n}$ is included in $\mathtt{conv}\left(\left\{\frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S)\right\}\right)$. Now, let us show that $\underline{\mathsf{MP}}(\rho)$ is smaller than any accumulation point of the infinite sequence of vectors $(\frac{1}{n} \cdot w(\rho \leq n)) \leq n$. Indeed, if x be an accumulation point of that sequence, then for all dimension i the sequence $\frac{1}{n} w_i(\rho \leq n)$ comes infinitely often arbitrarily close to x_i. This implies that $\liminf \frac{1}{n} w_i(\rho \leq n)$ is smaller than x_i for all dimensions i. Therefore $\underline{\mathsf{MP}}(\rho) \leq x$ and $\underline{\mathsf{MP}}(\rho) \in \downarrow \{x\} \subseteq \downarrow \mathtt{conv}\left(\left\{\frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S)\right\}\right).$ $\qquad\square$

Note that it is not always the case that $\underline{\mathsf{MP}}(\rho)$ is in the convex hull of $\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\}$. The example Fig. 4 shows that the downward closure operator is necessary. In this example, the sequence of vectors $(1,0)^{2^1} \cdot (0,1)^{2^2} \cdot (1,0)^{2^3} \cdot (0,1)^{2^4} \cdots$ which can be obtained with a path ρ that cycles on state s_1 is such that $\underline{\mathsf{MP}}(\rho) = (0,0)$
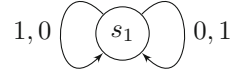

Fig. 4. Example of a game where $\underline{\mathsf{MP}}(\rho)$ does not belong to $\mathrm{conv}\left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\}\right)$ for all path ρ.

which is not in the convex hull of $(1,0)$ and $(0,1)$ (this convex hull is the set of points $(t, 1-t)$ with $t \in [\![0,1]\!]$).

Lemma 3. *For all $\sigma_\forall \in \mathbb{M}$, for all $s \in \mathsf{States}$, for all $S \in \mathit{SCC}(s, \sigma_\forall)$, for all $v \in \mathbf{conv}\left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\}\right)$, there exists an infinite path ρ of $\mathcal{G}(\sigma_\forall)$ starting from s such that $\underline{\mathsf{MP}}(\rho) = v$.*

Proof. Let $\{c_1, c_2, \ldots, c_n\}$ be a set of simple cycles in S such that $v = \sum_{i=1}^n \lambda_i \frac{1}{|c_i|} \cdot w(c_i)$, with $\sum_{i=1}^n \lambda_i = 1$, and $\lambda_i \in [0,1]$, i.e. v is a linear combination of the average weights of the simple cycles. It is proved in [15, Lemma 11], that we can build a path ρ that starts in s, reaches the SCC S and then cycles within S between the simple cycles $c_1, c_2, \ldots, c_n$ in such a way that the $\underline{\mathsf{MP}}$ on each dimension j, $1 \le j \le d$ is equal to $v_j = \sum_{i=1}^n \lambda_i \frac{1}{|c_i|} \cdot w(c_i)_j$. ∎

Characterizing the Feasible Thresholds. As `Adam` can play optimally with memoryless strategies, the set of feasible thresholds that `Eve` can force is obtained by considering the intersection of all the sets of thresholds that she can enforce against those memoryless strategies of `Adam`.

Theorem 4. *Let $\mathcal{G}$ be a game and s a state of $\mathcal{G}$:*

$$\mathbf{value}(\mathcal{G}, s) = \bigcap_{\sigma_\forall \in \mathbb{M}} \bigcup_{S \in \mathit{SCC}(s, \sigma_\forall)} \downarrow \mathbf{conv}\left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\}\right)$$

Proof. For the left to right inclusion. Assume that $\sigma_\exists$ is a winning strategy of `Eve` for the threshold v. For all memoryless strategies $\sigma_\forall \in \mathbb{M}$ of `Adam`, we have that $\rho = \mathsf{Outcome}(s, \sigma_\exists, \sigma_\forall)$ is such that $\underline{\mathsf{MP}}(\rho) \ge v$. By Lemma 2, $\underline{\mathsf{MP}}(\rho)$ belongs to the set $\downarrow \mathbf{conv}\left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\}\right)$, and as this set is downward closed it contains v.

For the right to left inclusion. Take any v in the set on the right. By Lemma 3, for all memoryless strategy $\sigma_\forall \in \mathbb{M}$ of `Adam`, we know that there exists an infinite path ρ starting from s in the graph $\mathcal{G}(\sigma_\forall)$ and such that $\underline{\mathsf{MP}}(\rho) = v$. This is equivalent to say that there exists a strategy $\sigma_\exists$ for `Eve` such that $\underline{\mathsf{MP}}(\mathsf{Outcome}(s, \sigma_\exists, \sigma_\forall)) = v$. So, this means that memoryless strategies of `Adam` cannot force from s an outcome with $\underline{\mathsf{MP}}$ which is not at least equal to v. As memoryless strategies of `Adam` are optimal, it means that `Adam` cannot obtain

from s an outcome with $\underline{\mathsf{MP}}$ which is not at least equal to v, no matter the strategy that he plays. As multidimensional mean-payoff games are determined [15], it means that Eve has a strategy to force outcomes from s with a $\underline{\mathsf{MP}}$ at least equal to v, which in turn implies that $v \in \mathtt{value}(\mathcal{G}, s)$. $\qquad\square$

Remark 2. As a corollary of Theorem 4, notice that the set $\mathtt{value}(\mathcal{G}, s)$ is closed.

Small Witnesses for Polyhedron Value Problem. We now show that if $\mathtt{value}(\mathcal{G}, s) \cap \mathtt{polyhedron}(\lambda) \neq \varnothing$, then there is a *witness* whose representation is polynomial.

Theorem 5. *There is a polynomial function P_4 such that, for all weighted game $\mathcal{G}$, for all states s, and system of linear inequations λ, if $\boldsymbol{value}(\mathcal{G}, s) \cap \boldsymbol{polyhedron}(\lambda) \neq \varnothing$ then there exists $x \in \mathbb{Q}^d$ such that:*

1. $x \in \boldsymbol{value}(\mathcal{G}, s) \cap \boldsymbol{polyhedron}(\lambda)$
2. $||x|| \leq P_4(d) \cdot (2 + \max\{||(a_j, b_j)|| \mid (a_j, b_j) \in \lambda\} + \log_2((W_\mathcal{G} + 1) \cdot (|\mathsf{States}| + 1)))$.

Proof. It follows from Theorem 4 that $\mathtt{value}(\mathcal{G}, s) \cap \mathtt{polyhedron}(\lambda) \neq \varnothing$ if, and only if, there is a function $f \colon \mathbb{M} \mapsto 2^{\mathsf{States}}$, such that $f(\sigma_\forall) \in \mathrm{SCC}(s, \sigma_\forall)$ for all strategy $\sigma_\forall$ and $\mathtt{polyhedron}(\lambda)$ intersects $\bigcap_{\sigma_\forall \in \mathbb{M}} \downarrow \mathtt{conv}\left(\left\{\frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(f(\sigma_\forall))\right\}\right)$. The values $\frac{1}{|c|} \cdot w(c)$ such that $c \in \mathbb{C}(S)$ for some SCC S, are such that their numerator is smaller in absolute value than $W_\mathcal{G} \cdot (|\mathsf{States}| + 1)$ and their denominator is smaller in absolute value than $|\mathsf{States}| + 1$. Let $a = (W_\mathcal{G} + 1) \cdot (|\mathsf{States}| + 1)$. We know by Theorem 3.3 that the set $\downarrow \mathtt{conv}\left(\left\{\frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S)\right\}\right)$ can be written as the intersection of half-spaces $H_1, \dots, H_k$ whose representation have size smaller than $P_2(d) \cdot (2 + \log_2(a))$. We conclude using Theorem 2 that there is a value $x \in \mathtt{value}(\mathcal{G}, s) \cap \mathtt{polyhedron}(\lambda)$ whose representation have size smaller than $d \cdot P_1(d) \cdot (1 + \max\{||H_i||, ||(a_j, b_j)|| \mid i \in [\![1, k]\!], (a_j, b_j) \in \lambda\})$ which is smaller than $d \cdot P_1(d) \cdot (1 + P_2(d) \cdot (2 + \log_2(a) + \max\{||(a_j, b_j)|| \mid (a_j, b_j) \in \lambda\}))$. We obtain the result for $P_4(d) = d \cdot P_1(d) \cdot (1 + P_2(d))$. $\qquad\square$

Based on this property, we design a non-deterministic algorithm and characterize the complexity of our decision problem.

Theorem 6. *The polyhedron value problem is $\Sigma_2 \mathsf{P}$-complete for mean-payoff inferior.*

Proof. **Easiness.** Based on Theorem 5, our algorithm is: 1. guess in polynomial time a value v; 2. check in deterministic polynomial time that it satisfies the set of linear equations λ (see e.g. [14, Theorem 3.3]); and 3. check in *non-deterministic* polynomial time that v belong to $\mathtt{value}(\mathcal{G}, s)$. This last check is based on the following result for the threshold problem: it is proved in [15, Theorem 7.2] that given a weighted game $\mathcal{G}$, a state s, and a threshold $v \in \mathbb{Q}^d$, the problem of deciding whether Eve has a winning strategy for the objective $\{\rho \mid \underline{\mathsf{MP}}(\rho) \geq v\}$ is coNP-complete. Our algorithm is thus in $\Sigma_2 \mathsf{P} = \mathsf{NP}^{\mathsf{NP}} = \mathsf{NP}^{\mathsf{coNP}}$.
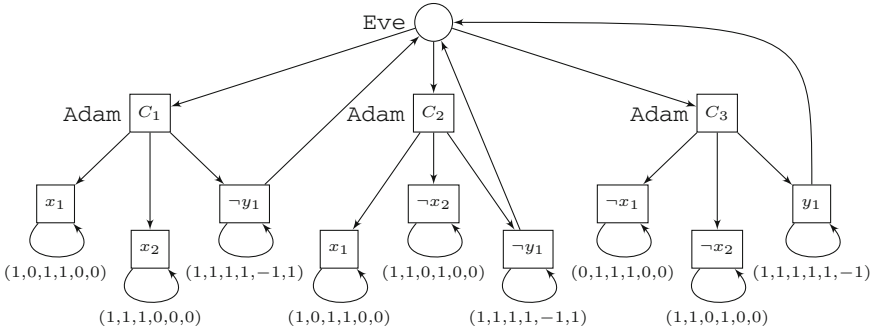
Fig. 5. Example of the encoding of QSAT_2 into the polyhedron value problem, for formula $\phi = \exists x_1.\exists x_2.\forall y_1.\ (x_1 \wedge x_2 \wedge \neg y_1) \vee (x_1 \wedge \neg x_2 \wedge \neg y_1) \vee (\neg x_1 \wedge \neg x_2 \wedge y_1)$. In a vector $(v_1, v_2, v_3, v_4, v_5, v_6)$, v_1 is associated to x_1, v_2 to $\neg x_1$, v_3 to x_2, v_4 to $\neg x_2$, v_5 to y_1, and v_6 to $\neg y_1$.

Hardness. We illustrate the reduction on an example. The full proof that the polyhedron value problem is $\Sigma_2\mathsf{P}$-hard can be found in the long version of this paper.

Consider a QSAT_2 formula: $\phi = \exists x_1.\ \cdots\exists x_n.\ \forall y_1.\ \cdots\forall y_m.\ C_1 \vee \cdots \vee C_p$ where each C_i is the conjunction of at most three literals $C_i = \ell_{i,1} \wedge \ell_{i,2} \wedge \ell_{i,3}$ and literals are of the form: x_j, $\neg x_j$, y_j, or $\neg y_j$. The construction of the game is illustrated in Fig. 5. There is one dimension for each literal of the formula. We consider the constraint λ that enforces than on each dimension associated to a literal y_j or $\neg y_j$ the mean-payoff should be greater than 0, and the sum on the dimension associated to x_j and $\neg x_j$ (for j fixed) should be 1. It is possible to show that the polyhedron value problem is true if, and only if, the formula ϕ is valid.

Intuitively, if ϕ is valid, there is a partial valuation of the x variables that makes the remainder of the formula hold. From this partial valuation, we define a vector v that is 1 on dimensions associated to x literals that are true and 0 on the other dimensions. Such a vector satisfies the constraints λ. For each memoryless strategy of Adam, we can construct a counter strategy of Eve that is winning, and this is enough to show that Eve has a winning strategy. To construct this strategy, first notice that if the strategy of Adam chooses one literal in some clause and its negation in another clause, Eve can win by alternating between the two (this ensures 0 for this literal and 1 for the others). We can now assume that the strategy of Adam defines a valuation for the literals y, by setting those that are reachable to true. Then because ϕ is valid, Eve can choose a clause that holds under the valuation that we defined. Then Adam will always chose a literal that is true under this valuation and this ensures a payoff above v.   □

Polynomial Time Algorithm for Fixed Number of Dimensions. We have seen in the previous paragraphs that the polyhedron value problem is $\Sigma_2\mathsf{P}$-complete. Now, we show that the problem has a much better worst-case

complexity for fixed number of dimensions and polynomially bounded weights, two hypotheses which are reasonable in practice.

Theorem 7. *The polyhedron value problem is solvable in polynomial time for mean-payoff inferior with fixed number of dimensions, polynomially bounded weights and a system a linear constraints with polynomially bounded numerators and denominators.*

Proof. Thanks to Theorem 5, for the polyhedron value problem, there are no more than $2^{P_4(d) \cdot (2+\max\{||(a,b)|| \mid (a,b) \in \lambda\} + \log_2((W_\mathcal{G}+1) \cdot (|\mathsf{States}|+1)))}$ candidate witnesses. This quantity is equal to $(4 \cdot 2^{\max\{||(a,b)|| \mid (a,b) \in \lambda\}} \cdot ((W_\mathcal{G}+1) \cdot (|\mathsf{States}|+1)))^{P_4(d)}$. Note that the size of equations in λ are logarithmic in the values of numerators and denominators that appear in it, so $2^{\max\{||(a,b)|| \mid (a,b) \in \lambda\}}$ is polynomial with respect to these values (but exponential with respect to d). The bound on the number of candidates is polynomial when the number of dimensions d is fixed and W is polynomially bounded. As the threshold problem for multidimensional weight mean-payoff games is solvable in polynomial time when the number of dimensions is fixed and the weights are polynomially bounded [8, Theorem 1], we obtain a polynomial time algorithm by simply testing all the polynomially many witnesses.   □

5   Constructing the Pareto Curve

Let $\mathcal{G}$ be a game of dimension d, we define the set of half-spaces $\mathcal{H}_\mathcal{G}$ and points $V_\mathcal{G}$ that are *relevant* for the representation of $\mathtt{value}(\mathcal{G}, s)$. $\mathcal{H}_\mathcal{G}$ and $V_\mathcal{G}$ are defined as the set of half-spaces and points with representation size bounded by $P_2(d) \cdot (2 + \log_2((W_\mathcal{G}+1) \cdot (|\mathsf{States}|+1)))$. The following lemma explains why those sets are relevant.

Lemma 4. *Let $\mathcal{G}$ be a game of dimension d and s a state of $\mathcal{G}$. The set $\boldsymbol{value}(\mathcal{G}, s)$ can be written as a finite union of polyhedra, each of them definable as the intersection of half-spaces $H_1, \ldots, H_k \in \mathcal{H}_\mathcal{G}$. Moreover if $H_1, \ldots, H_k$ are half-spaces of $\mathcal{H}_\mathcal{G}$ and $\cap_{1 \leq j \leq k} H_j \neq \varnothing$ then the intersection $\cap_{1 \leq j \leq k} H_j$ contains a point of $V_\mathcal{G}$.*

Equivalence Classes. We say that two points x and y are equivalent with respect to the set of half-spaces $\mathcal{H}$, written $x \sim_\mathcal{H} y$, if they satisfy the same set of equations and inequations defined by $\mathcal{H}$. Formally $x \sim_\mathcal{H} y$ if for all $H \in \mathcal{H}$, $x \in H \Leftrightarrow y \in H$ and $x \in \mathtt{hplane}(H) \Leftrightarrow y \in \mathtt{hplane}(H)$. Given a point x, we write $[x]_\mathcal{H} = \{y \mid x \sim_\mathcal{H} y\}$ the equivalence class containing x. These equivalence classes are known in geometry as *cells* [13]. We write $C(\mathcal{H})$ the set of cells defined by $\mathcal{H}$. The following lemma, says that cells which intersect $\mathtt{value}(\mathcal{G}, s)$ are included in it.

Lemma 5. *Let $c \in C(\mathcal{H}_\mathcal{G})$ be a cell, $c \cap \mathtt{value}(\mathcal{G}, s) \neq \varnothing$ if, and only if, $c \subseteq \mathtt{value}(\mathcal{G}, s)$.*

From this we deduce a method to compute a representation of the set $\mathtt{value}(\mathcal{G}, s)$ as a finite union of cells in $C(\mathcal{H}_\mathcal{G})$. Given a tuple of at most $d + 1$ points $X \subseteq V_\mathcal{G}$ with $|X| \leq d + 1$, we consider the geometrical center $b(X) = \sum_{x \in X} \frac{1}{d+1} \cdot x$. We write $B(V_\mathcal{G})$ the set of all these point, it contains at most $|V_\mathcal{G}|^{d+1}$ points. The following lemma states that $\mathtt{value}(\mathcal{G}, s)$ can be represented as the union of all cells that contain a point in $B(V_\mathcal{G})$ which is in $\mathtt{value}(\mathcal{G}, s)$. Note that the fact that we do not need many points to cover each cell is coherent with Buck's theorem [5,13].

Lemma 6. *Let $\mathcal{G}$ be a game of dimension d, s a state of $\mathcal{G}$, We have that:*

$$[-W_\mathcal{G}, W_\mathcal{G}]^d \cap \mathtt{value}(\mathcal{G}, s) = [-W_\mathcal{G}, W_\mathcal{G}]^d \cap \bigcup_{x \in B(V_\mathcal{G}) \cap \mathtt{value}(\mathcal{G}, s)} [x]_{\mathcal{H}_\mathcal{G}}$$

As a corollary, we obtain an effective procedure to compute a representation of the set $\mathtt{value}(\mathcal{G}, s)$. The complexity of this procedure is given in the following theorem both for the general case, and for fixed number of dimensions and polynomially bounded weights.

Theorem 8. *There is a deterministic exponential time algorithm that given a game $\mathcal{G}$ and a state s, constructs a effective representation of $[-W_\mathcal{G}, W_\mathcal{G}]^d \cap \mathtt{value}(\mathcal{G}, s)$ as a union of cells. Moreover, when the dimension d is fixed and weights are polynomially bounded in the size of $\mathcal{G}$, then the algorithm works in deterministic polynomial time.*

Proof. The algorithm is based on the result of Lemma 6. We enumerate all subsets of $d + 1$ points in $V_\mathcal{G}$. Because of their size, the number of points in $V_\mathcal{G}$ is bounded by $2^{P_4(d) \cdot (2 + \log_2((W_\mathcal{G}+1) \cdot (|\mathsf{States}|+1)))}$ which equals $(4 \cdot (W_\mathcal{G} + 1) \cdot (|\mathsf{States}| + 1))^{P_4(d)}$. The number of subsets of $d + 1$ points is thus bounded by $(4 \cdot (W_\mathcal{G} + 1) \cdot (|\mathsf{States}| + 1))^{P_4(d) \cdot (d+1)}$. Note that it is exponential in general, but with polynomially bounded weights, $W_\mathcal{G}$ is polynomial in the sizes of the input, so that with d fixed this number of subsets is polynomially bounded.

Now for each of these subsets, we compute the geometrical center x, and test whether it is in $\mathtt{value}(\mathcal{G}, s)$. Thanks to [8, Theorem 1], there is an algorithm that works in time $O(|\mathsf{States}|^2 \cdot |\mathsf{Edges}| \cdot d \cdot W_\mathcal{G} \cdot (d \cdot |\mathsf{States}| \cdot W_\mathcal{G})^{d^2 + 2 \cdot d + 1})$ to determine whether a point is in $\mathtt{value}(\mathcal{G}, s)$. This is exponential in general, but polynomial when the number of dimension is fixed and weights are polynomially bounded.

Then, to determine the cell corresponding to the geometrical center x, we test for each $H \in \mathcal{H}_\mathcal{G}$ whether $x \in H$: the intersection of the half-spaces that contain x and the complement of those that do not contain is equal to the cell containing x. Since the sizes of the half-spaces in $\mathcal{H}$ are bounded by $P_2(d) \cdot (2 + \log_2((W_\mathcal{G} + 1) \cdot (|\mathsf{States}| + 1)))$, we can test that x belongs to one of them in polynomial time and there are no more than $(4 \cdot (W_\mathcal{G} + 1) \cdot (|\mathsf{States}| + 1))^{P_2(d)}$ such half-spaces. Therefore testing all half-spaces can be done in exponential time in general, and in polynomial time with fixed dimension and polynomially bounded weights. □

Pareto Curve. The Pareto curve is composed of the maximal points in $[-W_{\mathcal{G}}, W_{\mathcal{G}}]^d \cap \texttt{value}(\mathcal{G}, s)$. To describe this curve, we need to refine the cells in $C(\mathcal{H}_{\mathcal{G}})$: we add to $\mathcal{H}_{\mathcal{G}}$ the half-spaces that are necessary to represent the downward closure of cells in $C(\mathcal{H}_{\mathcal{G}})$ (details can be found in the long version of this paper).

Theorem 9. *There is a deterministic exponential algorithm, that given a game $\mathcal{G}$ and a state s, computes an effective representation of $PO(\mathcal{G}, s)$ as a union of cells. Moreover, when the dimension d is fixed and the weights are polynomially bounded then the algorithm works in deterministic polynomial time.*

Proof (Sketch). The algorithm works by computing a representation of $\texttt{value}(\mathcal{G}, s)$ as a union of cells. Then, for each of these cells we check that there is no cell above by using the downward closure operator: this is where refining the cells is required. The number of those cells is exponential so these checks can be done in exponential time. Moreover in the case where the dimension is fixed and weights are polynomially bounded, this number is polynomial so the algorithm works in polynomial time. □

6   General Case

We now consider the general case in which the average of dimensions in $I \subset [\![1, d]\!]$ are defined using lim inf and the average of dimensions in $J \subseteq [\![1, d]\!]$ are defined using lim sup. We give a characterization of the feasible thresholds as we did in Theorem 4. While the main ideas are similar, the characterization here is substantially more complicated and relies on a notion of subgame defined as follows. A *subarena for* ***Eve*** is a tuple $\langle \mathsf{States}'_\exists, \mathsf{States}'_\forall, \mathsf{Edges}' \rangle$ with $\mathsf{States}' \subseteq \mathsf{States}$, $\mathsf{Edges}' \subseteq \mathsf{Edges}$ and such that $\forall s \in \mathsf{States}'_\forall. \ (s, s') \in \mathsf{Edges} \Rightarrow (s, s') \in \mathsf{Edges}'$ (i.e. it does not restrict actions of $\mathtt{Adam}$). The game $\langle \mathcal{A}', w', I', J' \rangle$ is a subgame for $\mathtt{Eve}$ of $\langle \mathcal{A}, w, I, J \rangle$ if $\mathcal{A}'$ is a subarena for $\mathtt{Eve}$ of $\mathcal{A}$ and $w' = w$, $I' = I$, and $J' = J$. We write $\mathsf{Sub}(\mathcal{G}, s)$ the set of subgames for $\mathtt{Eve}$ which contain the state s.

Theorem 10. *Let $\mathcal{G}$ be a weighted game and s a state of $\mathcal{G}$, then $\texttt{value}(\mathcal{G}, s)$ equals:*

$$\bigcup_{\mathcal{G}' \in \mathsf{Sub}(\mathcal{G}, s)} \bigcap_{s' \in \mathsf{States}'} \uparrow_J \left(\bigcap_{\sigma_\forall \in \mathbb{M}} \bigcup_{S' \in \mathcal{SCC}(s', \mathcal{G}'(\sigma_\forall))} \downarrow \texttt{conv}\left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S') \right\} \right) \right)$$

where $\uparrow_J X = \{ x \in \mathbb{R}^d \mid \forall j \in J. \ \exists x' \in X. \ \forall i \in I \cup \{j\}. \ x_i = x'_i \}$.

Example 1. Consider the example of Fig. 6. We choose $J = \{1, 2\}$ and $I = \{3\}$, i.e. we consider the limit superior of the weights for the two first coordinate and the limit inferior for the last one. There is only one strategy of the adversary and one strongly connected component in this game. There are two simple cycles
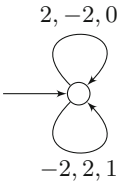
Fig. 6. A one-state 3-dimensional mean-payoff game, controlled by Eve. We refer to the 3 dimensions by x, y and z respectively.
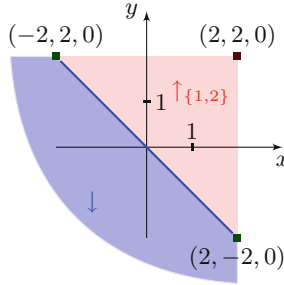
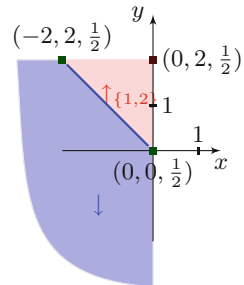**Fig. 7.** Pareto optimum for $z = 0$.

Fig. 8. Pareto optimum for $z = \frac{1}{2}$.

and their weight are $(2, -2, 0)$ and $(-2, 2, 1)$. We represented in Figs. 7 and 8 the feasible thresholds we can ensure with $z = 0$ and $z = \frac{1}{2}$.

For $z = 0$ the line segment between $(-2, 2, 0)$ and $(2, -2, 0)$ is below the convex hull of $(2, -2, 0)$ and $(-2, 2, 1)$. The downward closure this segment is the area that is below and left of this segment. The operator $\uparrow_{\{1,2\}}$ gives the whole area below of $(2, 2, 0)$ which is the Pareto optimum for $z = 0$. For $z = 1$ only $(-2, 2, 1)$ is below the weight of a simple cycle therefore it will be the Pareto optimum for $z = 1$. The convex hull of $(-2, 2, 1)$ and $(2, -2, 0)$ is above the plane $z = \frac{1}{2}$ for coordinates of x and y between $(0, 0)$ and $(-2, 2)$. The operator $\uparrow_{\{1,2\}}$ will give the whole area below $(0, 2, \frac{1}{2})$ which is the Pareto optimum for $z = 0$.

Thanks to the characterization of Theorem 10, we can express the value problem in terms of intersection of convex sets with a small description and using techniques similar to the ones used in the case of limit inferior we can show the following:

Theorem 11. *The polyhedron value problem is $\Sigma_2 P$-complete.*

The algorithm uses Theorem 10 and relies on the same principle as for $\lim \inf$.

References

1. Alur, R., Degorre, A., Maler, O., Weiss, G.: On omega-languages defined by mean-payoff conditions. In: de Alfaro, L. (ed.) FOSSACS 2009. LNCS, vol. 5504, pp. 333–347. Springer, Heidelberg (2009)
2. Bouyer, P., Brenguier, R., Markey, N., Ummels, M.: Concurrent games with ordered objectives. In: Birkedal, L. (ed.) FOSSACS 2012. LNCS, vol. 7213, pp. 301–315. Springer, Heidelberg (2012)

3. Brenguier, R., Raskin, J.-F.: Optimal values of multidimensional mean-payoff games. Research report, Université Libre de Bruxelles (U.L.B.), Belgium, October 2014. https://hal.archives-ouvertes.fr/hal-00977352
4. Brim, L., Chaloupka, J., Doyen, L., Gentilini, R., Raskin, J.-F.: Faster algorithms for mean-payoff games. Formal Methods Syst. Des. **38**(2), 97–118 (2011)
5. Buck, R.: Partition of space. Am. Math. Monthly **50**, 541–544 (1943)
6. 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)
7. Chatterjee, K., Majumdar, R., Henzinger, T.A.: Markov decision processes with multiple objectives. In: Durand, B., Thomas, W. (eds.) STACS 2006. LNCS, vol. 3884, pp. 325–336. Springer, Heidelberg (2006)
8. Chatterjee, K., Velner, Y.: Hyperplane separation technique for multidimensional mean-payoff games. In: D'Argenio, P.R., Melgratti, H. (eds.) CONCUR 2013 – Concurrency Theory. LNCS, vol. 8052, pp. 500–515. Springer, Heidelberg (2013)
9. Matoušek, J.: Lectures on Discrete Geometry, vol. 212. Springer, New York (2002)
10. Papadimitriou, C.H.: On the complexity of integer programming. J. ACM (JACM) **28**(4), 765–768 (1981)
11. Papadimitriou, C.H., Yannakakis, M.: On the approximability of trade-offs and optimal access of web sources. In: 41st Annual Symposium on Foundations of Computer Science, Proceedings, pp. 86–92. IEEE (2000)
12. Pnueli, A., Rosner, R.: On the synthesis of a reactive module. In: POPL 1989, pp. 179–190. ACM Press (1989)
13. Sack, J.-R., Urrutia, J.: Handbook of Computational Geometry. Elsevier, Amsterdam (1999)
14. Schrijver, A.: Theory of Linear and Integer Programming. Wiley, Chichester (1998)
15. Velner, Y., Chatterjee, K., Doyen, L., Henzinger, T.A., Rabinovich, A., Raskin, J.-F.: The complexity of multi-mean-payoff and multi-energy games. CoRR, abs/1209.3234 (2012)
16. Zwick, U., Paterson, M.: The complexity of mean payoff games on graphs. TCS **158**(1), 343–359 (1996)