



HAL
open science

Optimal Values of Multidimensional Mean-Payoff Games

Romain Brenguier, Jean-François Raskin

► **To cite this version:**

Romain Brenguier, Jean-François Raskin. Optimal Values of Multidimensional Mean-Payoff Games. [Research Report] Université Libre de Bruxelles (U.L.B.), Belgium. 2014. hal-00977352v4

HAL Id: hal-00977352

<https://hal.science/hal-00977352v4>

Submitted on 18 Oct 2014 (v4), last revised 22 May 2015 (v6)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimal Values of Multidimensional Mean-Payoff Games^{*}

Romain Brenguier, Jean-François Raskin

Université Libre de Bruxelles (U.L.B.), Belgium

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 classically called the *Pareto curve*, a classical tool to analyze *trade-offs*. Indeed, as weights are given as vectors in multiple dimensions, there can be incomparable such thresholds, and even an infinite number of incomparable ones. Our main results are as follow. First, we study the geometry of this set and show that it can be effectively represented as a finite union of convex sets. Second, we study the computational complexity of natural associated decision problems. In particular, we show that we can decide in $\Sigma_2\text{-P}$ if this set intersects a convex set of points defined by linear inequations. We also show that this problem is both NP-hard and coNP-hard. Third, we show that the Pareto curve can be approximated in polynomial time for fixed number of dimensions and unary encoding of weights.

1 Introduction

Two-player zero-sum games played on graphs are adequate models for open reactive systems [11], 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 (modelled as the winning condition in the game) against any behaviours of the environment. Recently, there has been a large effort to study quantitative extensions of those graph games, see e.g. [3]. 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. you 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 [13], the *threshold problem* for multi-dimensional mean-payoff games is studied, i.e. given a d -dimensional value vector $v \in \mathbb{Z}^d$, 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 can be incomparable thresholds that **Eve** is able to enforce, and even an infinite number of

^{*} Work supported by ERC Starting Grant inVEST (279499).

incomparable ones. The set of maximal thresholds that Eve can enforce is classically called the *Pareto curve*, it is a classical tool to analyze *trade-offs*. Another application of the Pareto curve is the study of multi-player games. For instance to compute Nash equilibria, a multiplayer game with mean-payoff objectives can be transformed into a multidimensional mean-payoff two-player game [2], and the Pareto curve of this multidimensional game is the most important step towards the computation of equilibria in the associated multi-player game. In this paper, we study the Pareto curve of thresholds that Eve can enforce in a multi-dimensional mean-payoff games. Our main contributions are as follows.

First, we characterize the geometry of this set and we show that this (usually infinite) set can be effectively represented as a (finite) union of convex sets defined by linear inequations. We obtain this results both for games where the mean-payoff is given dimension by dimension using \liminf (Theorem 8), and for a mixture of \liminf and \limsup (Theorem 13). As an application, because the Pareto curve is effectively representable as a finite union of convex sets, we can then optimize linear functions by calls to linear programming.

Second, we study the computational complexity of natural associated decision problems. In particular, we show how to decide in Σ_2 -P if the set of thresholds that Eve can enforce intersects a convex set of points defined by linear inequations. To obtain this result, we need to study finely the linear inequations used to characterise the Pareto curve. We show that, even if there are possibly an exponential number of convex sets, each of them being defined by potentially an exponential number of 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), are polynomial in the size of the game (Theorem 3). Additionally, we show that if the set has a non-empty intersection with a polyhedron defined by linear inequations then there is a polynomial size witness that can be check in coNP (Theorem 11). Then, we show that it is unlikely to improve substantially on this complexity as the non emptiness problem is both NP -hard and coNP -hard (Theorem 12).

Third, we show that it is possible to approximate the Pareto curve in pseudo-polynomial time for fixed number of dimensions both for notions of relative and absolute approximations (Theorem 17)¹. Those results are of practical relevance as the number of dimension while multiple is often quite low. For the absolute approximation, we show that the pseudo-polynomial time algorithm cannot be turn into a fully polynomial time algorithm without obtaining a polynomial time solution for 1-dimension mean-payoff games (Theorem 18), a long standing open question in the field [14].

Structure of the paper Sect. 2 provides the definition of the problems that we want to solve. Sect. 3 establishes some general results complexity results for manipulating polyhedra. Sect. 4 solves the \liminf case and Sect. 5 the general

¹ We thank Yaron Velner for pointing out his recent algorithm for solving the threshold problem [4]. This algorithm is instrumental to the pseudo-polynomial time algorithm that we describe in section 6

problem in which \liminf and \limsup are mixed. Sect. 6 provides results about the approximation of the Pareto curve.

2 Definitions

We define weighted arenas for players **Eve** and **Adam**.

Definition 1 (Arena). An arena \mathcal{A} is a tuple $\langle \text{Stat}_\exists, \text{Stat}_\forall, \text{Edg} \rangle$, where:

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

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

We write $\rho_{\geq n}$ for the suffix of ρ after $\rho_{\leq n-1}$, i.e. $\rho_n \cdot \rho_{n+1} \cdots$ and $\rho_{[m,n]}$ for $(\rho_{\leq n})_{\geq m}$.

A *history* h of the arena \mathcal{A} is a (finite) prefix of a play, i.e. an element of $\text{Stat}^* \cdot \text{Stat}$. We write h_i the i -th state of h , starting from 0, i.e. $h = h_0 \cdot h_1 \cdots h_n$. The length $|h|$ of such an history is $n + 1$. We write $\text{last}(h)$ the last state of h , i.e. $h_{|h|-1}$.

Let \mathcal{A} be an arena, a *strategy* for **Eve** maps histories ending in a state of Stat_\exists to a successor of that state. Formally it is a function $\sigma_\exists: \text{Stat}^* \cdot \text{Stat}_\exists \rightarrow \text{Stat}$, such that for all history h and state s , $(s, \sigma_\exists(h \cdot s)) \in \text{Edg}$. Similarly, a *strategy* for **Adam** is a function $\sigma_\forall: \text{Stat}^* \cdot \text{Stat}_\forall \rightarrow \text{Act}$, such that for all for all history h and state s , $(s, \sigma_\forall(h \cdot s)) \in \text{Edg}$. A strategy σ_\forall is *memoryless* if for every history h and h' , and all state 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 σ_\exists a strategy for **Eve**, a play ρ is *compatible* with the strategy σ_\exists if, for all $k \geq 0$, if $\rho_k \in \text{Stat}_\exists$ then $\rho_{k+1} = \sigma_\exists(\rho_{\leq k})$. We write $\text{Out}_{\mathcal{A}}(s, \sigma_\exists)$ for the set of plays in \mathcal{A} that are compatible with strategy σ_\exists and have initial state s (i.e. ρ such that $\rho_0 = s$). These plays are called *outcomes* of σ_\exists from s . We simply write $\text{Out}(s, \sigma_\exists)$ when \mathcal{A} is clear from context. The set of outcomes $\text{Out}_{\mathcal{A}}(s, \sigma_\forall)$ of a strategy of **Adam** is defined symmetrically.

Definition 2. A weighted game $\mathcal{G} = \langle \mathcal{A}, w, I, J \rangle$ is an arena \mathcal{A} equipped with a weight function $w: \text{Edg} \mapsto \mathbb{Z}^d$, and a partition $I \uplus J = \llbracket 1, d \rrbracket$.

Given a weight function w , we write w_i for the projection to the i -th dimension of the function w . The *mean-payoff inferior* and *mean-payoff superior* over

dimension i of a play ρ are given by:

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

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

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 game, s a state of \mathcal{G} , and $v \in \mathbb{R}^d$, we say that a strategy σ_{\exists} *ensures thresholds* v from state s if for all outcomes $\rho \in \text{Out}_{\mathcal{A}}(s, \sigma_{\exists})$, for all dimensions $i \in I$, $\underline{\text{MP}}_i(\rho) \geq v_i$, and for all dimensions $j \in J$, $\overline{\text{MP}}_j(\rho) \geq v_j$.

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. A threshold $v \in \mathbb{R}^d$ is said *Pareto optimal* from s if there is a strategy σ_{\exists} that ensures threshold v from s and all $v' \neq v$ which can be ensured by some σ'_{\exists} from s are such that $v'_i < v_i$ on some dimension i . We write $\text{PO}(\mathcal{G}, s)$ the set of Pareto optimal thresholds in game \mathcal{G} from state s . We refer to this set as the *Pareto curve* of the game. Our goal is to compute a representation of this curve. We also consider the set of *feasible thresholds* $\text{value}(\mathcal{G}, s)$ that contains all the thresholds that **Eve** can ensure and is in fact the downward closure of $\text{PO}(\mathcal{G}, s)$.

Linear inequations Let $a \in \mathbb{R}^d$ be a vector in d dimensions. The associated *linear function* $\alpha_a: \mathbb{R}^d \mapsto \mathbb{R}$ is the function $\alpha_a(x) = \sum_{i \in \llbracket 1, d \rrbracket} a_i \cdot x_i$ that computes the weighted sum relative to a . A *linear inequation* is a pair (a, b) where $a \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ and $b \in \mathbb{R}$. The *half-space* satisfying (a, b) is the set $\frac{1}{2}\text{space}(a, b) = \{x \in \mathbb{R}^d \mid \alpha_a(x) \geq b\}$. The *hyperplane* defined by (a, b) is the set $\text{hplane}(a, b) = \{x \in \mathbb{R}^d \mid \alpha_a(x) = b\}$. If $H = \frac{1}{2}\text{space}(a, b)$ is a half-space, we sometimes write $\text{hplane}(H)$ for the *associated hyperplane* $\text{hplane}(a, b)$. A *system of linear inequations* is a tuple $\lambda = ((a_1, b_1), \dots, (a_l, b_l))$ of linear inequations. The *polyhedron* generated by λ is the set $\text{polyhedron}(\lambda) = \bigcap_{k=1}^l \frac{1}{2}\text{space}(a_k, b_k)$.

A natural problem, is to try to optimize the threshold we can ensure with respect to a linear function $\alpha: \mathbb{R}^d \mapsto \mathbb{R}$. Given $v' \in \mathbb{R}_+^d$, we are looking for a strategy σ_{\exists} which ensures a threshold $v \in \mathbb{R}^d$, and such that there is no σ'_{\exists} which ensures a threshold v' , 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 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 σ_{\exists} and a value $v \in \text{polyhedron}(\lambda)$ such that σ_{\exists} ensures v . Note that this is equivalent to ask whether $\text{polyhedron}(\lambda)$ intersects $\text{value}(\mathcal{G}, s)$.

Remark 1. Other works ([13,4] 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 both NP-hard and coNP-hard while the 0-value problem is coNP-complete [13].

An example Assume that we have some resources R_1, \dots, R_n that are shared among d agents A_1, \dots, 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 spend 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 Figure 1.

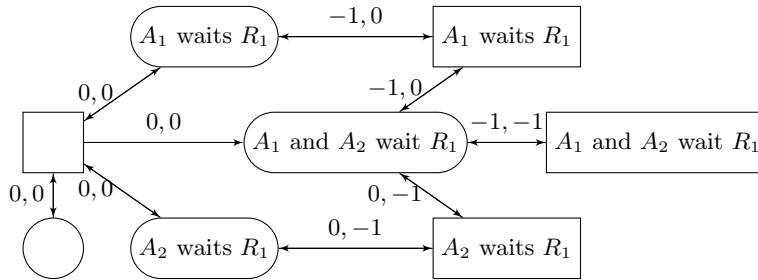


Figure 1. A two-dimensional mean-payoff game. Rounded states belong to the controller and rectangles to the environment.

On each dimension the average correspond to the opposite of the average time spend waiting by the agents. 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*. We represented in Figure 2 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.

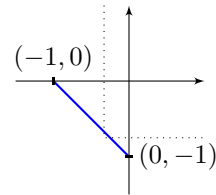


Figure 2. Pareto curve of the game of Figure 1.

3 Geometrical Notions

Since our typical reader may not be familiar with all the geometrical notions we will use, we summarize in this section useful notions and properties related to convex sets that will be useful to build our characterization of the sets $\text{PO}(\mathcal{G}, s)$ and $\text{value}(\mathcal{G}, s)$. For introduction to discrete geometry we refer the interested reader to [9]. We were not able to find in the literature on discrete geometry all the results that we needed for our characterization of the geometry of those two sets.²

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 $r = \frac{p}{q}$ 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 $v = (r_1, \dots, r_d)$ is $\|v\| = d + \sum_{i \in \llbracket 1, d \rrbracket} \|r_i\|$. The size of a matrix $A = (a_{ij})_{1 \leq i \leq d, 1 \leq j \leq n}$ is $\|A\| = m \cdot n + \sum_{i, j \in \llbracket 1, d \rrbracket \times \llbracket 1, n \rrbracket} \|a_{ij}\|$. The size of an equation (a, b) is $\|(a, b)\| = \|a\| + \|b\|$ and the size of a system of equation $\lambda = ((a_1, b_1), \dots, (a_n, b_n))$ is $\|\lambda\| = \sum_{i \in \llbracket 1, n \rrbracket} \|(a_i, b_i)\|$.

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 $H_F = \frac{1}{2} \text{space}(a_F, b_F)$ such that $P \cap \text{hplane}(a_F, b_F) = F$ and $P \subseteq H_F$. We will write $\mathcal{F}(P)$ for such a set.

A *affine subspace* is a set of the form $x + L$ with $x \in \mathbb{R}^d$ and L is a linear subspace of \mathbb{R}^d (i.e. a subset closed under addition of vectors and under multiplication by real numbers). The *affine hull* of a set $X \subseteq \mathbb{R}^d$ is the intersection of all affine subspaces of \mathbb{R}^d containing X , it is written $\text{aff}(X)$. It can be described as the set of linear combinations of points of X [9]: $\text{aff}(X) = \{\sum_{x \in X} t_x \cdot x \mid \forall x \in X. t_x \in \mathbb{R}\}$. The *convex hull* of a set of points $X \subseteq \mathbb{R}^d$ is the set $\text{conv}(X) = \{\sum_{x \in X} t_x \cdot x \mid \forall x \in X. t_x \in [0, 1] \wedge \sum_{x \in X} t_x = 1\}$. The *downward closure* of a set of points $X \subseteq \mathbb{R}^d$, is the set $\downarrow X = \{x \mid \exists x' \in X. \forall i \in \llbracket 1, n \rrbracket. x_i \leq x'_i\}$.

3.1 From the point representation to the half-space representation

If X is a finite set of points, the convex hull $P = \text{conv}(X)$ 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. We recall that given a polytope P and $\mathcal{F}(P)$ be complete set of

² If some results established in this section are easy consequences of known results in the literature, we would be happy to get relevant pointers.

facet-defining half-spaces for P , then P is the intersection of $\mathbf{aff}(P)$ with all half-spaces in $\mathcal{F}(P)$ [6, Thm. 3.(b)].

Theorem 1 ([6, Thm. 3.(b)]). *Let P be a polytope. Let $\mathcal{F}(P)$ be complete set of facet-defining half-spaces for P , then P is the intersection of $\mathbf{aff}(P)$ with all half-spaces in $\mathcal{F}(P)$.*

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 show that it is possible in Theorem 3, its proof (given in appendix) is based on the following basic property, see e.g. [12], that states that systems of linear equations that have solutions, have one of polynomial size.

Theorem 2 ([12, Corollary 3.2b]). *There exists a polynomial function P_1 , such that if the system $Ax = b$ of rational linear equations has a solution, it has one of size bounded by $P_1(\|A\|, \|b\|)$.*

We also show that the representation in term of half-spaces of the downward closure of a convex hull is of polynomial size. Proofs can be found in the appendix.

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

1. *given a finite set of points V , there are $k \leq |V|^d$ half-spaces H_1, \dots, H_k whose inequations have size smaller than $P_2(\max\{\|v\| \mid v \in V\}, d)$ and such that $\bigcap_{i \in [1, k]} H_i = \mathbf{conv}(V)$.*
2. *given a polytope (i.e a bounded polyhedron) defined by its set of facet-defining half-spaces $\mathcal{F} = \{\frac{1}{2} \mathbf{space}(a_1, b_1), \dots, \frac{1}{2} \mathbf{space}(a_k, b_k)\}$, it can be written as the convex hull $\mathbf{conv}(V)$ of a finite set V of points which all have size bounded by $P_3(\max\{\|a_i, b_i\| \mid 1 \leq i \leq k\}, d)$.*
3. *given a finite set of points V , there is a finite collection of half-spaces H_1, \dots, H_k which are described by inequations of size smaller than $P_2(\max\{\|v\| \mid v \in V\}, d)$, and such that $\bigcap_{i \in [1, k]} H_i = \downarrow \mathbf{conv}(V)$.*

Theorem 4. *There is a polynomial function P_2 such that given a finite set of points V , there are $k \leq |V|^d$ half-spaces H_1, \dots, H_k whose inequations have size smaller than $P_2(\max\{\|v\| \mid v \in V\}, d)$ and such that $\bigcap_{i \in [1, k]} H_i = \mathbf{conv}(V)$.*

Proof. Facets of $\mathbf{conv}(V)$ are given by d' linearly independent vertices of V where d' is the dimension of $\mathbf{conv}(V)$.

We first prove the case $d' = d$. Let f be a facet of $\mathbf{conv}(V)$ given by v_1, \dots, v_d which are d linearly independent points. We show how to obtain an equation of the half-space H^f defining this facet. The equation

$$\begin{pmatrix} v_1^T \\ \vdots \\ v_d^T \end{pmatrix} A = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

has a solution and Thm. 2 guarantees that we can find a solution of size bounded by $P_1(d^2 + \sum_{i \in \llbracket 1, d \rrbracket} \|v_i\|, 2 \cdot d)$. The equation $(A^T, 1)$ defines an hyperplane h . Moreover, for all v_j with $1 \leq j \leq k$, $A^T v_j = (v_j^T A)^T = 1$. This means that h contains all v_j and therefore either $\frac{1}{2}\mathbf{space}(A^T, 1)$ or $\frac{1}{2}\mathbf{space}(-A^T, -1)$ is defining the facet given by v_1, \dots, v_d . This gives us a description of H^f of size bounded by $P_2(\max\{\|v\| \mid v \in V\}, d) = P_1(d^2 + d \cdot \max\{\|v\| \mid v \in V\}, 2 \cdot d)$. Since a polytope is the intersection of its facet-defining half-spaces, $\mathbf{conv}(V)$ is the intersection of the H^f with f facet of $\mathbf{conv}(V)$.

We now assume that $d' < d$ and that the property holds in a space of dimension $d-1$. We look for half-spaces defining the affine subspace $A = \mathbf{aff}(\mathbf{conv}(V))$ of dimension d' containing $\mathbf{conv}(V)$. We take $d' + 1$ linearly independent points in V , which we write $v_1, \dots, v_{d'+1}$. Since $d' + 1 \leq d$ the equation:

$$\begin{pmatrix} v_1^T \\ \vdots \\ v_{d'+1}^T \end{pmatrix} B = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

has a solution, and using Thm. 2, there is a solution B whose size is bounded by $P_1(d'^2 + \sum_{i \in \llbracket 1, d' \rrbracket} \|v_i\|, 2 \cdot d')$. The equation $(B^T, 1)$ defines an affine subspace of dimension $d-1$ containing $\mathbf{conv}(V)$, so we can apply the induction hypothesis inside it. We obtain $H'_1, \dots, H'_{k'}$ half-spaces of A whose equations have size smaller than $P_2(\max\{\|v\| \mid v \in V\}, d')$ and such that $A \cap \bigcap_{i \in \llbracket 1, k' \rrbracket} H'_i = \mathbf{conv}(V)$. The equation defining $H'_1, \dots, H'_{k'}$ also define half-spaces of \mathbb{R}^d . The convex hull of V is then the intersection of $H'_1, \dots, H'_{k'}$ and the half-spaces given by $(B^T, 1)$ and $(-B^T, -1)$. The sizes of all the corresponding equations are bounded by $P_2(\max\{\|v\| \mid v \in V\}, d)$.

3.2 From the half-space representation to the point representation

Theorem 5. *There is a polynomial function P_3 such that a polytope (i.e a bounded polyhedron) given by its set of facet-defining half-spaces $\mathcal{F} = \frac{1}{2}\mathbf{space}(a_1, b_1), \dots, \frac{1}{2}\mathbf{space}(a_k, b_k)$, can be written as the convex hull $\mathbf{conv}(V)$ of a finite set V of points which all have size bounded by $P_3(\max\{\|a_i, b_i\| \mid 1 \leq i \leq k\}, d)$.*

Proof. Let P be the polytope $\bigcap_{1 \leq i \leq k} \frac{1}{2}\mathbf{space}(a_i, b_i)$ and d' be the dimension of $\mathbf{aff}(P)$. By [9, Prop. 5.3.2 (i)], the extremal points of a polytope are its vertices (i.e. faces of dimension 0), and a polytope is the convex hull of its vertices. Vertices are obtained by the intersection of d' hyperplane corresponding to facet-defining half-spaces. Therefore, for each vertice v , there is system of equations of size d' whose equations are given by a subset of the $(a_i, b_i)_{1 \leq i \leq k}$, such that v is the unique solution. Thanks to Thm. 2, this solution can be written with size bounded by $P_1(d'^2 + d \cdot \max\{\|a_i\| \mid i \in \llbracket 1, k \rrbracket\}, d \cdot (1 + \max\{\|b_i\| \mid i \in \llbracket 1, k \rrbracket\}))$. Taking V the set of vertices of P and considering the polynomial function $P_3(X, d) = P_1(d^2 + d \cdot X, d \cdot (1 + X))$, we obtain the desired result.

3.3 Downward closure

Lemma 1. *Operator \downarrow is monotonic, i.e. for any set X and Y such that $X \subseteq Y$, we have that $\downarrow X \subseteq \downarrow Y$.*

Proof. Assume $X \subseteq Y$. Let $x \in \downarrow X$, there is $x' \in X$ such that for all i , $x'_i \geq x_i$. As $X \subseteq Y$, $x' \in Y$, and therefore $x \in \downarrow Y$.

Lemma 2. *Operator \downarrow is compatible with union, i.e. for any set X and Y : $(\downarrow X) \cup (\downarrow Y) = \downarrow (X \cup Y)$.*

Proof. To show $(\downarrow X) \cup (\downarrow Y) \subseteq \downarrow (X \cup Y)$ We use the monotonicity property (Lem. 1): since $X \subseteq X \cup Y$, $\downarrow X \subseteq \downarrow (X \cup Y)$.

In the other direction, let $x \in \downarrow (X \cup Y)$, there exists $x' \in X \cup Y$ such that $x \leq x'$. If $x' \in X$ then $x \in \downarrow X$ and otherwise $x \in \downarrow Y$.

Equations of the downward closure of a polyhedron Our goal is from the representation in term of half-spaces of a polyhedron, to obtain a representation of its downward closure. We will show that the downward closure of convex hull can also be represented by relatively small (i.e. polynomial) equations. We will prove several lemmas to obtain the full characterization of Corollary 1. But we first introduce some notations.

Let X be a set of points and $J \subseteq \llbracket 1, d \rrbracket$, we write:

- $\pi_J(X)$ for the set $\{x' \mid \exists x \in X. \forall i \in \llbracket 1, d \rrbracket \setminus J. x'_i = x_i \text{ and } \forall j \in J. x'_j = 0\}$,
- $\downarrow_J(X)$ for the set $\{x' \mid \exists x \in X. \forall i \in \llbracket 1, d \rrbracket \setminus J. x'_i = x_i \text{ and } \forall j \in J. x'_j \leq x_j\}$,
- $\uparrow_J(X)$ for the set $\{x' \mid \exists x \in X. \forall i \in \llbracket 1, d \rrbracket \setminus J. x'_i = x_i \text{ and } \forall j \in J. x'_j \in \mathbb{R}\}$;

We also write $\mathbf{1}_i$ for the point that has coordinate 1 on dimension i and 0 on the other dimensions. We write C_J for the cone $\sum_{j \in J} \mathbb{R}_+ \cdot \mathbf{1}_j$. If P is a polyhedron and $\mathcal{F}(P)$ a set of facet-defining half-spaces of P in the subspace $\mathbf{aff}(P)$, we write $\mathcal{F}_-^J(P) \subseteq \mathcal{F}(P)$ the half-spaces H of $\mathcal{F}(P)$ that contain the direction $-C_J$, i.e. $H - C_J = H$.

Lemma 3. *Let P be a polyhedron, $J \subseteq \llbracket 1, d \rrbracket$, and $\mathcal{F}(\pi_J(P))$ be a set of half-spaces of $\mathbf{aff}(\pi_J(P))$. We have that:*

$$\uparrow_J P = \bigcap_{H \in \mathcal{F}(\pi_J(P))} H + \mathbb{R}^J$$

where \mathbb{R}^J denotes the subspace $\sum_{j \in J} \mathbb{R} \cdot \mathbf{1}_j$.

Proof.

$$\begin{aligned} x \in \uparrow_J P &\Leftrightarrow \exists x' \in P. \pi_J(x) = \pi_J(x') \\ &\Leftrightarrow \pi_J(x) \in \bigcap_{H \in \mathcal{F}(\pi_J(P))} H \\ &\Leftrightarrow \pi_J(x) + \sum_{j \in J} \mathbb{R} \cdot \mathbf{1}_j \subseteq \bigcap_{H \in \mathcal{F}(\pi_J(P))} H + \sum_{j \in J} \mathbb{R} \cdot \mathbf{1}_j \\ &\Leftrightarrow x \in \bigcap_{H \in \mathcal{F}(\pi_J(P))} H + \mathbb{R}^J \end{aligned}$$

□

We show how to obtain facet-defining half-spaces of the downward closure of P with respect to one dimension.

Lemma 4. *Let P be a polyhedron, $\mathcal{F}(P)$ be a set of facet-defining half-spaces of P and $j \in \llbracket 1, d \rrbracket$. We have:*

$$\downarrow_j P = \uparrow_j P \cap \bigcap_{H \in \mathcal{F}_-^j(P)} H$$

Proof. $\boxed{\subseteq}$ Let $x \in \downarrow_j P$. We have that $x \in \uparrow_j P$. There is $x' \in P$ such that $x = x' - \lambda \cdot \mathbf{1}_j$ with $\lambda \geq 0$. Let $f \in \mathcal{F}_-^j$, since $x' \in H_f$ we have that $x \in H_f - \mathbb{R}^+ \cdot \mathbf{1}_j = H_f$. Therefore $x \in \uparrow_j P \cap \bigcap_{f \in \mathcal{F}_-^j} H_f$.

$\boxed{\supseteq}$ Let $x \in \uparrow_j P \cap \bigcap_{f \in \mathcal{F}_-^j} H_f$. There exists $x' \in P$ such that $\pi_{\{j\}}(x) = \pi_{\{j\}}(x')$. Consider the line segment $[x, x']$, it does not intersect any $\mathbf{hplane}(H_f)$ with $f \in \mathcal{F}_-^j$ since both x and x' are inside these half-spaces.

If $[x, x']$ does not intersect any $\mathbf{hplane}(H_f)$ where f is a facet of P then x is in the intersection of these half-spaces, and using Thm. 1, this means that $x \in P$.

Otherwise $[x, x']$ intersects some $\mathbf{hplane}(H_f)$ with f facet of P which is not in \mathcal{F}_-^j . Since H_f contains x' but not x , and these two only differs by their j coordinates, this means that $x_j \leq x'_j$ and therefore $x \in \downarrow_j P$. □

We now generalize the previous lemma to the downward closure with respect to several dimensions.

Lemma 5.

$$\downarrow_K P = \bigcap_{J \subseteq K} \bigcap_{H \in \mathcal{F}_-^{K \setminus J}(\pi_J(P))} H + \mathbb{R}^J$$

with the convention that $\pi_\emptyset(P) = P$, $\mathcal{F}_-^\emptyset(P) = \mathcal{F}(P)$ and $B_\emptyset = \mathbf{0}$.

Proof. We prove the result by induction over the size of K . The case where $|K| = 1$ is a direct consequence of Lem. 3 and 4.

Assume the equality holds for some set K . Consider a dimension $j \notin K$.

$$\pi_{\{j\}}(\downarrow_K P) = \downarrow_K \pi_{\{j\}}(P) = \bigcap_{J \subseteq K} \bigcap_{H \in \mathcal{F}_-^{K \setminus J}(\pi_{J \cup \{j\}}(P))} H + B_J$$

So we can chose:

$$\mathcal{F}(\pi_{\{j\}}(\downarrow_K P)) = \bigcup_{J \subseteq K} \{H + \mathbb{R}^J \mid H \in \mathcal{F}_-^{K \setminus J}(\pi_{J \cup \{j\}}(P))\} \quad (1)$$

We also have by induction hypothesis:

$$\mathcal{F}(\downarrow_K P) = \bigcup_{J \subseteq K} \{H + \mathbb{R}^J \mid H \in \mathcal{F}_-^{K \setminus J}(\pi_{\{j\}}(P))\}$$

$$\mathcal{F}_-^j(\downarrow_K P) = \bigcup_{J \subseteq K} \{H + \mathbb{R}^J \mid H \in \mathcal{F}_-^{K \cup \{j\} \setminus J}(\pi_{\{j\}}(P))\} \quad (2)$$

Using Lem. 3 and 4:

$$\begin{aligned} \downarrow_{K \cup \{j\}} P &= \bigcap_{H \in \mathcal{F}(\pi_{\{j\}}(\downarrow_K P))} H + B_j \cap \bigcap_{H \in \mathcal{F}_-^j(\downarrow_K P)} H \\ &= \bigcap_{J \subseteq K} \bigcap_{H \in \mathcal{F}_-^{K \setminus J}(\pi_{J \cup \{j\}}(P))} H + \mathbb{R}^J + B_j \cap \bigcap_{H \in \mathcal{F}_-^j(\downarrow_K P)} H \quad \text{By (1)} \\ &= \bigcap_{J \subseteq K} \bigcap_{H \in \mathcal{F}_-^{K \setminus J}(\pi_{J \cup \{j\}}(P))} H + B_{J \cup \{j\}} \cap \bigcap_{J \subseteq K} \bigcap_{H \in \mathcal{F}_-^{K \cup \{j\} \setminus J}(\pi_{\{j\}}(P))} H + \mathbb{R}^J \quad \text{By (2)} \\ &= \bigcap_{J \subseteq K \cup \{j\}} \bigcap_{H \in \mathcal{F}_-^{K \cup \{j\} \setminus J}(\pi_J(P))} H + B_J \end{aligned}$$

Which proves the property for $K \cup \{j\}$ and concludes the induction. \square

From this lemma we deduce a representation of the downward closure of a polyhedron in term of half-spaces.

Corollary 1. *If P is a polyhedron, then:*

$$\downarrow P = \bigcap_{J \subseteq \llbracket 1, d \rrbracket} \bigcap_{H \in \mathcal{F}_-^{\llbracket 1, d \rrbracket \setminus J}(\pi_J(P))} H + \mathbb{R}^J$$

Theorem 6. *There is a polynomial function P_2 such that given a finite set of points V , there is a finite collection of half-spaces H_1, \dots, H_k which are described by equations of size smaller than $P_2(\max\{\|v\| \mid v \in V\}, d)$, and such that $\bigcap_{i \in \llbracket 1, k \rrbracket} H_i = \downarrow \text{conv}(V)$.*

Proof. We will in fact use the same polynomial function P_2 than for Thm. 3.

We use Corollary 1, the collection H_1, \dots, H_k is given by the half-spaces of the form $H + \mathbb{R}^J$ with $J \subseteq \llbracket 1, d \rrbracket$, and $H \in \mathcal{F}_-^{\llbracket 1, d \rrbracket \setminus J}(\pi_J(\text{conv}(V)))$. We show that these half-spaces are described by small equations.

We fix such J and $H \in \mathcal{F}_-^{\llbracket 1, d \rrbracket \setminus J}(\pi_J(\text{conv}(V)))$. The polytope $\pi_J(\text{conv}(V))$ can be obtained as the convex hull of $\pi_J(V)$. The set $\pi_J(V)$ is a finite set of points whose sizes are smaller than $\max\{\|v\| \mid v \in V\}$. Using Thm. 3.4, half-spaces of $\mathcal{F}(\pi_J(\text{conv}(V)))$ can be described with size smaller than $P_2(\max\{\|v\| \mid v \in \pi_J(V)\}, d - |J|)$ and thus with size smaller than $P_2(\max\{\|v\| \mid v \in V\}, d)$. The set of facet $\mathcal{F}_-^{\llbracket 1, d \rrbracket \setminus J}(\pi_J(\text{conv}(V)))$ is include in $\mathcal{F}(\pi_J(\text{conv}(V)))$ and therefore the size of the equation of H is also bounded by $P_2(\max\{\|v\| \mid v \in V\}, d)$.

Now let (A, b) be such an equation of H , it also defines a half-space H' of \mathbb{R}^d which obviously contains H . We define A' to be such that for all $i \in \llbracket 1, d \rrbracket \setminus J$, $A'_i = A_i$ and for all $j \in J$, $A'_j = 0$. Note that the size of A' is bounded by that of A and therefore the size of the equation is bounded by $P_2(\max\{\|v\| \mid v \in V\}, d)$. If $A' = (0, \dots, 0)$ this contradicts the fact that H is defining a facet of

$\text{aff}(\pi_J(\text{conv}(V)))$. Otherwise the equation (A', b) defines a half-space which we write $H' = \frac{1}{2}\text{space}(A', b)$. We have that $H' \cap \pi_J(\mathbb{R}^d) = H$ since all points of π_J have coordinate on J equal to 0 and only coefficient on these dimensions differ between A and A' . Moreover it contains the direction \mathbb{R}^J since for all $j \in J$, $A' \cdot \mathbf{1}_j = 0$. Therefore H' is equal to $H + \mathbb{R}^J$. This shows that $\text{conv}(V)$ can be described by equation that have size smaller than $P_2(\max\{\|v\| \mid v \in V\}, d)$.

Small witnesses of large systems of inequations

Our algorithm to decide the polyhedron value problem needs to guess solutions of large systems of inequations. We need to ensure that the solution that it guesses is of polynomial size. Intuitively, our result (Theorem 7) says that if a system of inequations as 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 Figure 3: in two dimensions, if a collection of half-spaces (i.e. half-planes here) intersect (green area in the picture), then either there is a point at the intersection of two boundary lines which is in the intersection, or one of these lines is included in the intersection.

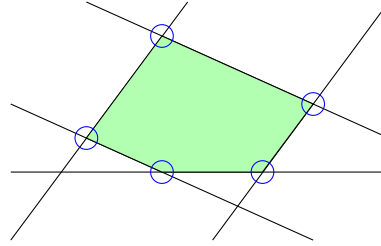


Figure 3. Illustration of Lem. 6 in the case of two dimensions: In this example, the possible witnesses of the property are circled.

Lemma 6. *Let H_1, \dots, H_n be n inequations of \mathbb{R}^d . If $\bigcap_{i=1}^n \frac{1}{2}\text{space}(H_i) \neq \emptyset$ then there are $k \leq d$ indexes i_1, \dots, i_d such that: $\bigcap_{j=1}^k \text{hplane}(H_{i_j}) \neq \emptyset$ and $\bigcap_{j=1}^k \text{hplane}(H_{i_j}) \subseteq \bigcap_{i=1}^n \frac{1}{2}\text{space}(H_i)$.*

Proof. We will prove by induction a slightly stronger version of the lemma as follows: let S be an affine subspace of \mathbb{R}^d of dimension k , with $1 \leq k \leq d$. If $S \cap \bigcap_{i=1}^n \frac{1}{2}\text{space}(H_i) \neq \emptyset$ then there are $k' \leq k$ indexes $i_1, \dots, i_{k'}$ such that:

$$S \cap \bigcap_{j=1}^{k'} \text{hplane}(H_{i_j}) \neq \emptyset \quad \text{and} \quad S \cap \bigcap_{j=1}^k \text{hplane}(H_{i_j}) \subseteq S \cap \bigcap_{i=1}^n \frac{1}{2}\text{space}(H_i).$$

The lemma correspond to the case where $S = \mathbb{R}^d$.

We show this property by induction over k . If $k = 0$, then if $S \cap \bigcap_{i=1}^n \frac{1}{2}\text{space}(H_i) \neq \emptyset$ then every $\frac{1}{2}\text{space}(H_i)$ contains S since it is reduced to one point. Therefore $S \cap \bigcap_{i=1}^0 \frac{1}{2}\text{space}(H_i) = S = S \cap \bigcap_{i=1}^n \frac{1}{2}\text{space}(H_i)$.

Assume now $k > 0$ and that the property holds for all affine subspace of dimension $k' < k$. We show the property for k by induction over n . If $n = 0$ the property obviously holds.

We now assume that the property holds for $n \geq 0$ and prove it for $n + 1$. We consider a half-space H_{n+1} .

- If $S \subseteq \mathbf{hplane}(H_{n+1})$ then we use the induction hypothesis for n : there are $k' \leq k$ indexes $i_1, \dots, i_{k'}$ such that $\bigcap_{j=1}^{k'} (\mathbf{hplane}(H_{i_j}) \cap S) \neq \emptyset$ and:

$$\begin{aligned} \bigcap_{j=1}^{k'} (\mathbf{hplane}(H_{i_j}) \cap S) &\subseteq S \cap \bigcap_{i=1}^n \frac{1}{2} \mathbf{space}(H_i) \\ &\subseteq S \cap \mathbf{hplane}(H_{n+1}) \cap \bigcap_{i=1}^n \frac{1}{2} \mathbf{space}(H_i) \\ &\subseteq S \cap \bigcap_{i=1}^{n+1} \frac{1}{2} \mathbf{space}(H_i). \end{aligned}$$

Thus the property holds for $n + 1$ in that case.

- Otherwise $\mathbf{hplane}(H_{n+1}) \cap S$ forms a affine subspace of dimension $k - 1$, to which we can apply the induction hypothesis.
 - If $S \cap \mathbf{hplane}(H_{n+1}) \cap \bigcap_{i=1}^n \frac{1}{2} \mathbf{space}(H_i) \neq \emptyset$ then there are $k' \leq k - 1$ indexes $i_1, \dots, i_{k'}$ such that $(\mathbf{hplane}(H_{n+1}) \cap S) \cap \bigcap_{j=1}^{k'} \mathbf{hplane}(H_{i_j}) \neq \emptyset$ and :

$$\begin{aligned} (\mathbf{hplane}(H_{n+1}) \cap S) \cap \bigcap_{j=1}^{k'} \mathbf{hplane}(H_{i_j}) &\subseteq S \cap \mathbf{hplane}(H_{n+1}) \cap \bigcap_{i=1}^n \frac{1}{2} \mathbf{space}(H_i) \\ S \cap \bigcap_{j=1}^{k'} \mathbf{hplane}(H_{i_j}) \cap \mathbf{hplane}(H_{n+1}) &\subseteq S \cap \bigcap_{i=1}^{n+1} \frac{1}{2} \mathbf{space}(H_i) \end{aligned}$$

Which shows the property.

- Otherwise $(\bigcap_{i=1}^n \frac{1}{2} \mathbf{space}(H_i) \cap S) \cap \mathbf{hplane}(H_{n+1}) = \emptyset$. Let $S' = \bigcap_{i=1}^n \frac{1}{2} \mathbf{space}(H_i) \cap S$, we show that it is either included in $\frac{1}{2} \mathbf{space}(H_{n+1})$ or it does not intersect it: assume there is one point $x \in S'$ inside $\frac{1}{2} \mathbf{space}(H_{n+1})$ and one point $x' \in S'$ outside of it, then $\alpha_a(x) \geq b$ and $\alpha_a(x') < b$. By continuity of linear functions, there would be $x'' \in [x, x']$ such that $\alpha_a(x'') = b$. Since S' is convex $x'' \in S'$, and therefore $x'' \in S' \cap \mathbf{hplane}(H_{n+1})$ which contradicts the hypothesis.

If S' is outside $\frac{1}{2} \mathbf{space}(H_{n+1})$ then the intersection $S \cap \bigcap_{i=1}^{n+1} \frac{1}{2} \mathbf{space}(H_i)$ is empty, so we have nothing to prove.

Otherwise, $\bigcap_{i=1}^n \frac{1}{2} \mathbf{space}(H_i) \cap S \subseteq \frac{1}{2} \mathbf{space}(H_{n+1})$. We apply the induction hypothesis over n : there are $k' \leq k$ indexes $i_1, \dots, i_{k'}$ such that $\bigcap_{j=1}^{k'} \mathbf{hplane}(H_{i_j}) \cap S \neq \emptyset$ and :

$$\bigcap_{j=1}^{k'} \mathbf{hplane}(H_{i_j}) \cap S \subseteq \bigcap_{i=1}^n \frac{1}{2} \mathbf{space}(H_i) \cap S \subseteq S \cap \bigcap_{i=1}^{n+1} \frac{1}{2} \mathbf{space}(H_i)$$

Which proves the result. \square

Thanks to this property we can deduce that we can find small solutions for system of inequations, independently of the number of inequations.

Theorem 7. *There is a polynomial function P_4 such that if the system of inequations $\lambda = ((a_1, b_1), \dots, (a_n, b_n))$ has a solution then there is a point x such that $\|x\| \leq P_4(\max\{\|(a_i, b_i)\| \mid i \in \llbracket 1, n \rrbracket\}, d)$ and $x \in \text{polyhedron}(\lambda)$.*

Proof. Let $\lambda = ((a_1, b_1), \dots, (a_n, b_n))$ be a system of inequations defining a non empty polyhedron. By Lem. 6, there is a set H_1, \dots, H_k subset of $\{\frac{1}{2}\text{space}(a_i, b_i) \mid i \in \llbracket 1, n \rrbracket\}$ with $k \leq d$ and such that: $\bigcap_{j=1}^k \text{hplane}(H_{i_j}) \neq \emptyset$ and $\bigcap_{j=1}^k \text{hplane}(H_{i_j}) \subseteq \bigcap_{i=1}^n \frac{1}{2}\text{space}(H_i)$. We can renumber the inequations such $(a_1, b_1), \dots, (a_k, b_k)$ are the equation associated with H_1, \dots, H_k respectively. Since $\bigcap_{j=1}^k \text{hplane}(H_{i_j}) \neq \emptyset$ the system of equation $((a_1, b_1) \dots (a_k, b_k))$ has a solution. Using Thm. 2 there is a solution x whose size is bounded by $P_1(k \cdot d + \sum_{i \in \llbracket 1, d \rrbracket} \|a_i\|, k + \sum_{i \in \llbracket 1, d \rrbracket} \|b_i\|)$ where P_1 is a polynomial function. Since $k \leq d$, the size of x can be bounded by $P_4(\max\{\|(a_i, b_i)\| \mid i \in \llbracket 1, n \rrbracket\}, d) = P_1(d^2 + d \cdot \max\{\|(a_i, b_i)\| \mid i \in \llbracket 1, n \rrbracket\}, d + d \cdot \max\{\|(a_i, b_i)\| \mid i \in \llbracket 1, n \rrbracket\})$. Since $\bigcap_{j=1}^k \text{hplane}(H_{i_j}) \subseteq \bigcap_{i=1}^n \frac{1}{2}\text{space}(H_i)$, the solution x is also inside the intersection of $\bigcap_{i=1}^n \frac{1}{2}\text{space}(H_i)$, and thus $x \in \text{polyhedron}(\lambda)$.

4 The limit inferior case

We first solve the case where the average of the weights are defined using limit inferior in all dimensions (i.e. $I = \llbracket 1, d \rrbracket$ and $J = \emptyset$). In this case, the set of thresholds that can be ensured by Eve is:

$$\text{value}(s) = \{v \in \mathbb{R}^d \mid \exists \sigma_{\exists}. \forall \rho \in \text{Out}(s, \sigma_{\exists}). \forall i \in \llbracket 1, d \rrbracket. \underline{\text{MP}}_i(\rho) \geq v_i\}.$$

To obtain a geometrical characterization of this set, we first show that the set of relevant thresholds when Adam plays according to a fixed memoryless strategy is a finite union of convex sets. Then, we show that the set $\text{value}(s)$ can be obtained as the intersection of those sets for all the memoryless strategies of Adam, as stated in Thm. 8. From this characterization, using the results of previous section, we deduce that if there is a solution to the polyhedron value problem, then there is one of small size. In turn, this allows us to define a $\Sigma_2\text{P}$ -algorithm for the polyhedron value problem. Finally, we show that this problem is both NP-hard and coNP-hard, so its complexity is unlikely to be substantially lower.

Adam plays optimally with memoryless strategies Memoryless strategies for Adam are important as we know that Adam can always play optimally with a memoryless strategy in the threshold problem as proved in [13], i.e. if he has strategy to prevent Eve from ensuring some threshold then he has a memoryless one. We characterize here what are the possible mean-payoffs of outcomes that Eve can obtain when Adam plays according to a fixed memoryless strategy.

4.1 Decomposition in Simple Cycles

Our analysis relies on the analysis of simple cycles. Let \mathcal{A} be an arena and $S \subseteq \text{Stat}$ a subset of nodes. A *simple cycle* within S is sequence $s_0 \cdot s_1 \cdots s_n$,

such that $s_0 = s_n$, $s_i \in S$, for all $0 \leq i \leq n$, 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 .

Our proof will use as a tool the decomposition of a play into *simple cycles*.

We recall the notion of *decomposition into simple cycles* of a play [7]. Every history h of a finite game can be uniquely decomposed into a sequence of simple cycles, except for a finite part. The decomposition process maintains a *stack*, $\mathbf{st}(h)$, of distinct states and moves. We write the stack content $s_1 \cdot s_2 \cdots s_n$ where s_1 is at the bottom of the stack and s_n the top. We use the notation $s \in \mathbf{st}(h)$ for $s \in \{s_1, s_2, \dots, s_n\}$. The *decomposition*, $\mathbf{dec}(h)$, is a sequence of simple cycles. We define $\mathbf{dec}(h)$ and $\mathbf{st}(h)$ inductively as follows:

- for the single state history s , $\mathbf{dec}(s) = \emptyset$ and $\mathbf{st}(s) = s$.
- let $h' = h \cdot s$ be a history.
 - If $s \in \mathbf{st}(h)$, and $\mathbf{st}(h) = \alpha \cdot s \cdot \beta$, then $\mathbf{st}(h') = \alpha \cdot s$ and $\mathbf{dec}(h') = \mathbf{dec}(h) \cdot (s \cdot \beta \cdot s)$.
 - else $\mathbf{dec}(h') = \mathbf{dec}(h)$, $\mathbf{st}(h') = \mathbf{st}(h) \cdot s$.

Note that the stack always contains distinct elements, therefore only simple cycles are added to the decomposition. The elements in the stack from the bottom to the top, form a history $s_0 \cdot s_1 \cdots s_n$, where $n + 1$ is the height of the stack.

The decomposition of a play is the limit of the decompositions of finite prefixes of the play. This is well defined for the distance $d(\rho, \rho') = 2^{\min\{i \mid \rho_i \neq \rho'_i\}}$ since if h is a prefix of h' then $\mathbf{dec}(h)$ is a prefix of $\mathbf{dec}(h')$.

Let $\sigma_{\forall} \in \mathbb{M}$ be a memoryless strategy, it defines a graph $\mathcal{G}(\sigma_{\forall}) = \langle \mathbf{Stat}, \mathbf{Edg}_{\sigma_{\forall}} \rangle$ where $\mathbf{Edg}_{\sigma_{\forall}} = \{(s, s') \in \mathbf{Edg} \mid s \in \mathbf{Stat}_{\exists} \vee (s \in \mathbf{Stat}_{\forall} \wedge \sigma_{\forall}(s) = s')\}$ which is a subgraph of the game arena. Let $\mathbf{SCC}(s, \sigma_{\forall})$ be the set of *strongly connected components* accessible from s in this subgraph $\mathcal{G}(\sigma_{\forall})$, that is the set of maximal subsets of nodes that are strongly connected in $\mathcal{G}(\sigma_{\forall})$.

The following lemma states that the mean-payoff of a run is a convex combination of the weights of the cycles of some strongly connected component.

Lemma 7. *Let ρ be a path in $\mathcal{G}(\sigma_{\forall})$. If $\mathbf{Inf}(\rho) \subseteq S$ for some $S \in \mathbf{SCC}(s, \sigma_{\forall})$, then $\mathbf{MP}(\rho) \in \downarrow \mathbf{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\} \right)$.*

Proof. A point x is an *accumulation point* of the sequence x_0, x_1, \dots if, for every open set containing x , there are infinitely many indices such that the corresponding elements of the sequence belong to the open set. It is mentioned as a remark in [1] that a run ρ will eventually come into the SCC S that it will never leave (i.e $\mathbf{Inf}(\rho) \subseteq S$) and the set of accumulation points of the sequence $(\frac{1}{n} \cdot w(\rho_{\leq n}))_{\leq n}$ is included in the convex hull of the simple cycles of the SCC S . We now show that $\mathbf{MP}(\rho)$ is smaller than any accumulation point of $(\frac{1}{n} \cdot w(\rho_{\leq n}))_{\leq n}$. Let x be an accumulation point of $(\frac{1}{n} \cdot w(\rho_{\leq n}))_{\leq n}$, 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 . This being true for all dimensions i , we have that $\mathbf{MP}(\rho) \leq x$. Therefore $\mathbf{MP}(\rho) \in \downarrow \{x\} \subseteq \downarrow \mathbf{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\} \right)$.

Note that we cannot show that $\underline{\text{MP}}(\rho)$ is in the convex hull of $\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\}$ in general, and we need the downward closure of this set instead. The example Figure 4 illustrates this. In this example, the path $(1, 0)^{2^1} \cdot (0, 1)^{2^2} \cdot (1, 0)^{2^3} \dots$ gives a limit inferior of 0 on both dimensions. However the simple cycles have weight $(1, 0)$ and $(0, 1)$ respectively, so the convex hull is the set of points $(t, 1 - t)$ with $t \in \llbracket 0, 1 \rrbracket$ which does not contain $(0, 0)$.

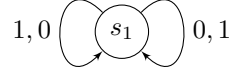


Figure 4. Example of a game where $\underline{\text{MP}}$ does not belong to $\text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\} \right)$.

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 memoryless strategies of Adam.

Theorem 8. *Let \mathcal{G} be a weighted game with $J = \emptyset$ and s a state of \mathcal{G} .*

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

Our proof will rely on this two lemmas:

Lemma 8 ([13, Lem. 10(2)]). *Let S be a strongly connected graph. If S does not have a non-negative multi-cycle (i.e. $C \in \mathbb{C}(S)$ and $\lambda \in \mathbb{N}^C$ such that $\sum_{c \in C} \lambda(c) \cdot w(c) \geq 0$), then there exists a constant $m_G \in \mathbb{N}$ and a real $c_G > 0$ such that for all history h in the graph S we have $\min\{w_i(h) \mid i \in I\} \leq m_G - c_G \cdot |h|$.*

Lemma 9 ([13, Lem. 11]). *Let σ_{\vee} be a strategy of Adam, and s be a state. If there is a strongly connected component S reachable from s in $\mathcal{G}(\sigma_{\vee})$ which has a non-negative multi-cycle, then Eve has a strategy to satisfy the mean-payoff-inf (i.e. $\forall i \in I \cup J. \forall \rho \in \text{Out}_s(\sigma_{\exists}). \underline{\text{MP}}_i(\rho) \geq 0$).*

Proof (of Thm. 8). $\boxed{\subseteq}$ Let $v \in \text{value}(s)$, then there is a strategy σ_{\exists} of Eve, such that for all memoryless strategies $\sigma_{\vee} \in \mathbb{M}$ of Adam, $\forall i \in \llbracket 1, d \rrbracket. \underline{\text{MP}}_i(\text{Out}(\sigma_{\exists}, \sigma_{\vee})) \geq v_i$. Let $\rho = \text{Out}(\sigma_{\exists}, \sigma_{\vee})$. We have that $\text{Inf}(\rho) \subseteq S$ for some $S \in \text{SCC}(\sigma_{\vee})$. Lem. 7 shows that $\underline{\text{MP}}(\rho) \in \downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\} \right)$ which implies that $v \in \downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\} \right)$.

$\boxed{\supseteq}$ Let $v \in \bigcap_{\sigma_{\vee} \in \mathbb{M}} \bigcup_{S \in \text{SCC}(\sigma_{\vee})} \downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\} \right)$. We will show that Eve has a winning strategy for v against all memoryless strategy of Adam and then use the fact that if Adam has a winning strategy then he has a memoryless one [13, Thm. 8].

We fix a memoryless strategy σ_{\vee} of Adam. We have that v is in $\downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\} \right)$ for some $S \in \text{SCC}(\sigma_{\vee})$.

We first reduce the problem to the case where $v = (0, \dots, 0)$ by subtracting v from all weights. We replace the weights of the game by w' such that $w'_i(s) = w_i(s) - v_i$. We have for all history h that if the average weight $\frac{w(h)}{|h|}$ is greater than v then $w'(\rho) \geq (0, \dots, 0)$. We know there exists some $v' \geq v$ in $\text{conv}\left(\left\{\frac{1}{|c|} \cdot w(c) \mid c \in \mathcal{C}(S)\right\}\right)$. We can write v' in the form $\sum_{k=1}^{|C|} \frac{\lambda_k}{|c_k|} \cdot w(c_k)$ with all $\lambda_k \in [0, 1]$ and $\sum_{k=1}^{|C|} \lambda_k = 1$. Hence there exists $\lambda_1, \dots, \lambda_{|C|}$ such that $\sum_{k=1}^{|C|} \frac{\lambda_k}{|c_k|} \cdot w'(c_k) \geq 0$.

The idea is to construct for all n a finite path of length greater than n that has a total weight by w' close to 0. Let n be an integer, there exist $p_k \in \mathbb{N}$ such that $|\frac{p_k}{n} - \lambda_k| \leq \frac{1}{n}$. Let $l_k = \prod_{k' \neq k} |c_{k'}|$, we construct the history h that follows $p_1 \cdot l_1$ times the cycle c_1 , then goes to the beginning of cycle c_2 with a path of length at most $|\text{Stat}|$ then follows $p_2 \cdot l_2$ times the cycle c_2 and so on until the $|C|$ -th cycle.

First, lets look at the length of this path.

$$\begin{aligned} \sum_{k=1}^{|C|} \frac{p_k}{n} &\geq \sum_{k=1}^{|C|} \lambda_k - \frac{|C|}{n} \\ \sum_{k=1}^{|C|} p_k &\geq n \sum_{k=1}^{|C|} \lambda_k - |C| = n - |C| \end{aligned}$$

Hence $|h| \geq n - |C|$, hence it is not bounded when n grows toward infinity.

Let W' be the maximum weight that appears in w' . On dimension i , the total weight of h is such that:

$$\begin{aligned} \sum_{k=1}^{|C|} p_k \cdot l_k \cdot w'_i(c_k) - |\text{Stat}| \cdot |C| \cdot |W'| &\leq w'_i(h) \\ \prod_{k=1}^{|C|} |c_k| \cdot \sum_{k=1}^{|C|} p_k \cdot \frac{w'_i(c_k)}{|c_k|} - |\text{Stat}| \cdot |C| \cdot |W'| &\leq w'_i(h) \\ \prod_{k=1}^{|C|} |c_k| \cdot \sum_{k=1}^{|C|} \left(\lambda_k - \frac{1}{n}\right) \cdot \frac{w'_i(c_k)}{|c_k|} - |\text{Stat}| \cdot |C| \cdot |W'| &\leq w'_i(h) \\ -\frac{1}{n} \cdot \prod_{k=1}^{|C|} |c_k| \cdot \sum_{k=1}^{|C|} \frac{w'_i(c_k)}{|c_k|} - |\text{Stat}| \cdot |C| \cdot |W'| &\leq w'_i(h) \end{aligned}$$

Hence $w'_i(h)$ is bounded on all dimension i when we make n grow.

By Lem. 10(2) of [13]³ this implies there is a non negative cycle in C for the weight w' . By Lem. 11 of [13], σ_\forall is not a winning strategy. We know thanks

³ Note that in [13] the weights are integer, which may not be the case for us. However the proof of Lem. 10(2) still works in the case of arbitrary reals.

to Theorem 8 of [13] that multi-mean-payoff games are determined under memoryless strategies of **Adam**. Therefore **Adam** has no winning strategy. Since these games are also determined [8], this means that **Eve** has a winning strategy. Therefore $v \in \text{value}(s)$. \square

Proof (Sketch⁴). The inclusion is a consequence of consequence of Lem. 7. Now in the other direction, let v be in the second set. Fixing a memoryless strategy σ_V of **Adam**, we have that v is smaller or equal to some linear combination of cycles in $\text{SCC}(s, \sigma_V)$. We construct a history h which approximates this linear combination by cycling an adequate number of time in cycles of $\text{SCC}(\sigma_V)$. By constructing histories h longer and longer we obtain a sequence of histories for which the sum of the weight $w - v$ is bounded below. Thanks to [13, Lem. 10(2)], this means there is a non negative cycle in C for the weight w' . By Lem. 11 of [13], σ_V is not a winning strategy. Thanks to Theorem 8 of [13], multi-mean-payoff games are determined under memoryless strategies of **Adam** and since these games are also determined [8], this means that **Eve** has a winning strategy for threshold v .

Simple solutions Let us now show that if there is a solution to the polyhedron value problem then there is a witness threshold which has coordinate of polynomial size. First note that solving this problem is equivalent to determine whether $(s) \cap \text{polyhedron}(\lambda) \neq \emptyset$.

Theorem 9. *There is a polynomial function P_5 such that for all weighted game \mathcal{G} and system of linear inequation λ , if $(s) \cap \text{polyhedron}(\lambda) \neq \emptyset$ then there is a point x in the intersection such that $\|x\| \leq P_5(\|\lambda\|, \|W\|, \|\text{Stat}\| + 1, d)$, where W is the maximal value appearing in the weights.*

Proof. Let λ be a system of linear inequations. It follows from Thm. 8 that $\text{value}(s) \cap \text{polyhedron}(\lambda) \neq \emptyset$ if, and only if, there is a function $f: \mathbb{M} \mapsto 2^{\text{Stat}}$, such that $f(\sigma_V) \in \text{SCC}(s, \sigma_V)$ for all strategy σ_V and $\text{polyhedron}(\lambda)$ intersects $\bigcap_{\sigma_V \in \mathbb{M}} \downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(f(\sigma_V)) \right\} \right)$. The points $\frac{1}{|c|} \cdot w(c)$ such that $c \in \mathbb{C}(S)$ for some strongly connected component S have size smaller than $d \cdot (\|W\| + \|\text{Stat}\| + 1) + \|\text{Stat}\| + 1$. As we saw in Thm. 3.6, the set $\downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\} \right)$ can be written as the intersection of half-spaces whose equations have polynomial size in that of the points of the form $\frac{1}{|c|} \cdot w(c)$. We conclude using Thm.7 that there is a point $x \in (s) \cap \text{polyhedron}(\lambda)$ whose size is bounded by $P_5(\|\lambda\|, \|W\|, \|\text{Stat}\|, d)$ which can be taken as $P_4(\|\lambda\| + d \cdot (\|W\| + \|\text{Stat}\| + 1) + \|\text{Stat}\| + 1, d)$. \square

To solve the polyhedron value problem, our algorithm will guess some point and check that it satisfies the linear equations and that it belong to **value**. For this last step, there exists a **coNP** algorithm as we recall here.

Theorem 10 ([13, Thm. 7.2]). *For multi-weighted games with objective $\{\rho \mid \text{MP}(\rho) \geq (0, \dots, 0)\}$, the problem of deciding whether a given state is winning for **Eve** is **coNP**-complete.*

⁴ The full proof can be found in the appendix

Given a vector v , we can subtract it from all the weights of the game \mathcal{G} and using the preceding algorithm, then ensuring $(0, \dots, 0)$ in this new game is equivalent to ensuring v in \mathcal{G} . We therefore have the following corollary.

Corollary 2. *Given a vector $v \in \mathbb{Q}^d$ and a state s , we can decide with a coNP algorithm whether $v \in \mathbf{value}(s)$.*

To summarize, our algorithm to decide polyhedron value problem which proceeds as follows: existentially guess a point $x \in [-W, W]^d$ such that $\|x\| \leq P_5(\|\lambda\|, \|W\|, \|n\|, d)$; if $x \in \mathbf{polyhedron}(\lambda)$ and $x \in \mathbf{value}(s)$ then answers yes and no otherwise. Its correctness holds from Thm. 9. The algorithm is in $\Sigma_2\mathbf{P} = \mathbf{NP}^{\mathbf{NP}} = \mathbf{NP}^{\mathbf{coNP}}$, since checking whether $x \in \mathbf{value}(s)$ can be done in coNP (by Corollary 2) and computing the intersection of hyperplanes can be done in polynomial time [12, Thm. 3.3]. This results in the following theorem.

Theorem 11. *The polyhedron value problem is in $\Sigma_2\mathbf{P}$ for mean-payoff inferior.*

Hardness We conclude this section by showing that it is unlikely to improve substantially on the complexity of our algorithm as we show that the polyhedron value problem is both NP-hard and coNP-hard.

Theorem 12. *The polyhedron value problem is both coNP-hard and NP-hard for limit inferior objectives, and for limit superior objectives.*

Proof. Concerning coNP-hardness, this is a consequence of the co-NP-hardness for the value $(0, \dots, 0)$ problem [13]. We can code the question whether $(0, \dots, 0)$ can be ensured by a polyhedron value problem by taking equations $\lambda = (\lambda_1, \dots, \lambda_d)$ where $\lambda_i = (\mathbf{1}_i, 0)$.

Let $\phi = \exists x_1. \exists x_2. \dots \exists x_n. C_1 \wedge \dots \wedge C_n$ be a formula, where $C_i = \ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}$ with $\ell_{i,j} \in \{x_k, \neg x_k \mid k \in \llbracket 1, n \rrbracket\}$. We have two dimensions for each variable x_i , so $d = 2 \cdot n$. We write 0_i for the vector v such that $v_i = 0$ and $v_j = 1$ for $j \neq i$. We define a family of vectors $v_{i,j}$ where if $\ell_{i,j} = x_k$ then $v_{i,j} = 0_{2 \cdot k + 1}$ and if $\ell_{i,j} = \neg x_k$ then $v_{i,j} = 0_{2 \cdot k}$. We have one initial state s_0 controlled by **Adam**, one state for each clause C_i controlled by **Eve**, and one state for each literal of the formula. In s_0 **Adam** chooses a clause C_i and then **Eve** chooses a literal $\ell_{i,j}$ of C_i . From there the weights are equal to $v_{i,j}$. We consider the constraints $\lambda_i = (v_{2 \cdot i} + v_{2 \cdot i + 1} \geq 1)$. The construction is illustrated in Figure 5. The correctness of the construction is proved in the appendix, here we will only illustrate it on an example. Let ξ be a valuation, we associate to it a vector v_ξ such that if $\xi(x_i) = \mathbf{true}$ then $(v_\xi)_{2 \cdot i} = 1$ and $(v_\xi)_{2 \cdot i + 1} = 0$; if $\xi(x_i) = \mathbf{false}$ then $(v_\xi)_{2 \cdot i} = 0$ and $(v_\xi)_{2 \cdot i + 1} = 1$. In the example of Figure 5, valuation $x_1 \rightarrow \mathbf{true}, x_2 \rightarrow \mathbf{false}, x_3 \rightarrow \mathbf{true}$ makes the formula true, the corresponding value is $v = (1, 0, 0, 1, 1, 0)$. Consider the strategy of **Eve** that in C_1 chooses x_1 , in C_2 chooses x_1 and in C_3 chooses $\neg x_2$. In all the cases this strategy ensures to **Eve** a payoff better than v . In contrast, valuation $x_1 \rightarrow \mathbf{false}, x_2 \rightarrow \mathbf{false}, x_3 \rightarrow \mathbf{true}$ does not make the formula true, the corresponding value is $v' = (0, 1, 0, 1, 1, 0)$, and if **Adam** chooses C_1 then **Eve** cannot ensure v' .

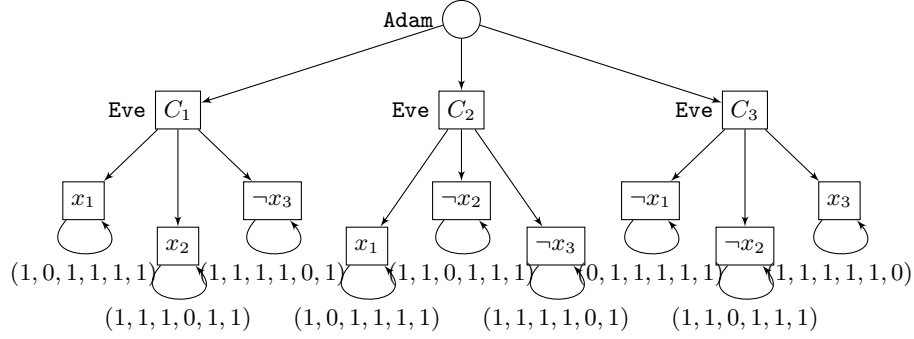


Figure 5. Example of the encoding of 3SAT into the polyhedron value problem, for formula $\phi = (x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$.

Proof (Continued). Let ξ be a valuation, we associate to it a vector v_ξ such that if $\xi(x_i) = \text{true}$ then $(v_\xi)_{2 \cdot i} = 1$ and $(v_\xi)_{2 \cdot i + 1} = 0$; if $\xi(x_i) = \text{false}$ then $(v_\xi)_{2 \cdot i} = 0$ and $(v_\xi)_{2 \cdot i + 1} = 1$. Assume ξ makes ϕ true, we will prove that **Eve** can ensure v_ξ . We consider a strategy for **Eve** that in each C_i chooses a state $\ell_{i,j}$ such that ξ makes $\ell_{i,j}$ true, this is possible because v_ξ makes ϕ true. If $\ell_{i,j} = x_k$, then $\xi(x_k) = \text{true}$ and $(v_\xi)_{2 \cdot k + 1} = 0$. The payoff is then $0_{2 \cdot k + 1}$ which is greater than v_ξ . Similarly, if $\ell_{i,j} = \neg x_k$, the payoff $0_{2 \cdot k}$ is greater than v_ξ . This shows that **Eve** has a strategy to ensure the threshold v_ξ . We conclude that if ϕ is satisfiable then **Eve** can ensure a threshold that satisfies the equations in λ .

Now, reciprocally, assume there is some v that satisfies λ and that can be ensured by **Eve**. For each $k \in \llbracket 1, n \rrbracket$, $v_{2 \cdot k} > 0$ or $v_{2 \cdot k + 1} > 0$ as v satisfies λ . We select a valuation ξ_v such that $\xi_v(x_k) = \text{true} \Leftrightarrow v_{2 \cdot k} > 0$. Assume **Eve** ensures v , we will prove that ξ_v makes ϕ true. From every C_i , she can find a $\ell_{i,j}$ such that $v_{i,j} \geq v$. If $\ell_{i,j} = x_k$, then $v_{i,j} = 0_{2 \cdot k + 1}$, hence $(v_\xi)_{2 \cdot k + 1} = 0$. Therefore $\xi_v(x_k) = \text{true}$. Similarly if $\ell_{i,j} = \neg x_k$, then $\xi_v(x_k) = \text{false}$. This means that in each C_i , we can find a literal made true by ξ_v and therefore ξ_v makes ϕ true.

This proves that ϕ is satisfiable if, and only if, **Eve** can ensure a threshold that satisfies the inequations in λ . \square

5 General case

We now consider the general case in which the average of dimensions in $I \subseteq \llbracket 1, d \rrbracket$ are defined using \liminf and the average of dimensions in $J \subseteq \llbracket 1, d \rrbracket$ are defined using \limsup . We give a characterization of the feasible thresholds as we did in Thm. 8. 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 $\langle \text{Stat}'_{\exists}, \text{Stat}'_{\forall}, \text{Edg}' \rangle$ with $\text{Stat}' \subseteq \text{Stat}$, $\text{Edg}' \subseteq \text{Edg}$ and such that $\forall s \in \text{Stat}'_{\forall}. (s, s') \in \text{Edg} \Rightarrow (s, s') \in \text{Edg}'$ (i.e. it does not restrict actions of **Adam**). The game $\langle \mathcal{A}', w', I', J' \rangle$ is a subgame for **Eve** of

$\langle \mathcal{A}, w, I, J \rangle$ if \mathcal{A}' is a subarena for Eve of \mathcal{A} and $w' = w$, $I' = I$, and $J' = J$. We write $\text{Sub}(\mathcal{G}, s)$ the set of subgame for Eve which contain the state s .

Theorem 13. *Let \mathcal{G} be a weighted game and s a state of \mathcal{G} , then:*

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

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

We recall here some properties of multi-mean-payoff games which were proved in [13].

Lemma 10 ([13, Lem. 17]).

$$\begin{aligned} & \forall s \in \text{Stat}. \exists \sigma_{\exists}. \forall \sigma_{\forall}. \forall i \in I. \underline{\text{MP}}_i(\text{Out}(s, \sigma_{\exists}, \sigma_{\forall})) \geq v_i \text{ and } \forall j \in J. \overline{\text{MP}}_j(\text{Out}(s, \sigma_{\exists}, \sigma_{\forall})) \geq v_j \\ \Leftrightarrow & \forall s \in \text{Stat}. \forall j \in J. \exists \sigma_{\exists}. \forall \sigma_{\forall}. \forall i \in I. \underline{\text{MP}}_i(\text{Out}(s, \sigma_{\exists}, \sigma_{\forall})) \geq v_i \text{ and } \overline{\text{MP}}_j(\text{Out}(s, \sigma_{\exists}, \sigma_{\forall})) \geq v_j. \end{aligned}$$

Lemma 11 ([13, Lem. 14]). *Let $j \in J$ and $s \in \text{Stat}$.*

$$\begin{aligned} & \exists \sigma_{\exists}. \forall \sigma_{\forall}. \forall i \in I. \underline{\text{MP}}_i(\text{Out}(s, \sigma_{\exists}, \sigma_{\forall})) \geq v_i \text{ and } \overline{\text{MP}}_j(\text{Out}(s, \sigma_{\exists}, \sigma_{\forall})) \geq v_j \\ \Leftrightarrow & \exists \sigma_{\exists}. \forall \sigma_{\forall}. \forall i \in I. \underline{\text{MP}}_i(\text{Out}(s, \sigma_{\exists}, \sigma_{\forall})) \geq v_i \text{ and } \underline{\text{MP}}_j(\text{Out}(s, \sigma_{\exists}, \sigma_{\forall})) \geq v_j. \end{aligned}$$

The proof of the theorem relies on the following lemma which is illustrated by Example 1. From this lemma, we prove inclusion in Thm. 13 by taking for \mathcal{G}' the restriction of \mathcal{G} to states accessible when Eve plays a strategy that ensures the given value. We prove the inclusion in the other direction by noticing that it is easier to win in \mathcal{G} than in a subgame \mathcal{G}' and conclude using Lem. 12.

Lemma 12. *Let \mathcal{G} be a weighted game with $J \neq \emptyset$, and s a state of \mathcal{G} , then:*

$$\bigcap_{s \in \text{Stat}} (s) = \bigcap_{s \in \text{Stat}} \uparrow_J \left(\bigcap_{\sigma_V \in \mathbb{M}} \bigcup_{S \in \text{SCC}(s, \sigma_V)} \downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\} \right) \right)$$

Proof. Let v be a point in $\text{value}(s)$ for all state s , then by definition: $\forall i \in I$, $\underline{\text{MP}}_i(\rho) \geq v_i$ and $\forall j \in J$, $\overline{\text{MP}}_j(\rho) \geq v_j$. Thanks to [13, Lem. 17], this is equivalent to $\forall s \in \text{Stat}. \forall j \in J. \exists \sigma_{\exists}. \forall \sigma_{\forall}. \forall i \in I. \underline{\text{MP}}_i(\rho) \geq v_i$ and $\overline{\text{MP}}_j(\rho) \geq v_j$. Then by [13, Lem. 14], this is also equivalent to $\forall s \in \text{Stat}. \forall j \in J. \exists \sigma_{\exists}. \forall \sigma_{\forall}. \forall i \in I. \underline{\text{MP}}_i(\rho) \geq v_i$ and $\underline{\text{MP}}_j(\rho) \geq v_j$. This means that $\forall s \in \text{Stat}, v \in \uparrow_J \text{value}(\mathcal{G}', s)$, where game \mathcal{G}' is obtained from \mathcal{G} by setting $I' = I \cup J$ and $J' = \emptyset$. We then obtain the result using 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 and their weight are $(2, -2, 0)$ and $(-2, 2, 1)$. We represented in Figure 8 and Figure 8 the the feasible thresholds we can ensure with $z = 0$ and $z = \frac{1}{2}$.

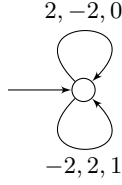


Figure 6. A one-state multidimensional mean-payoff game, controlled by Eve.

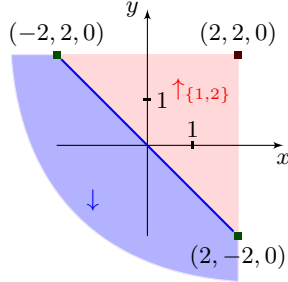


Figure 7. Pareto optimum for $z = 0$.

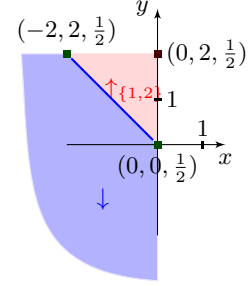


Figure 8. Pareto optimum for $z = \frac{1}{2}$.

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 $z = 0$.

Proof (of Thm. 13). We first prove inclusion. Let $v \in \text{value}(s)$, and σ_{\exists} a strategy which ensures it. Let S be the set of states that is reachable from s by following σ_{\exists} . We have $\{s\} \subseteq S \subseteq \text{Stat}$. We also have $v \in \bigcap_{s' \in S} \text{value}(s')$, since otherwise v would not be ensured by σ_{\exists} . Restricting the game to vertices of S defines a subgame \mathcal{G}' , since only edges of Eve have been removed, it is a subgame for Eve. In a subgame for Eve the values can only be lower than in the original game. Using Lem. 12 on this game, we obtain that v belongs to $\bigcap_{s \in S} \uparrow_J \bigcap_{\sigma_{\forall} \in \mathbb{M}} \bigcup_{S \in \text{SCC}(s, \sigma_{\forall})} \downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S) \right\} \right)$, which shows the inclusion.

Now, in the other direction, let v be an element of:

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

Let \mathcal{G}' be a subgame for Eve such that $v \in \bigcap_{s \in \text{Stat}'} \uparrow_J \bigcap_{\sigma_{\forall} \in \mathbb{M}} \bigcup_{S' \in \text{SCC}(s, \mathcal{G}'(\sigma_{\forall}))} \downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S') \right\} \right)$. We use Lem. 12 on this subgame \mathcal{G}' , to obtain:

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

Since the actions of **Adam** have not been restricted, it is harder to ensure a given threshold in \mathcal{G}' than in \mathcal{G} . Therefore as v is ensured in \mathcal{G}' from all states of S , it can also be ensured in \mathcal{G} from all states of Stat' , which shows the reverse inclusion.

5.1 Expressing thresholds as a union of convex sets

Thanks to the previous characterization, we can express the value problem in terms of intersection of convex sets with a small description.

We first prove the following lemma that will allow use to reorder union and \uparrow operators in the equation of Thm. 13.

Lemma 13. *Operator \uparrow (defined in appendix 3.3) is compatible with union, i.e. for all $J \subseteq \llbracket 1, d \rrbracket$ and all set X and Y : $(\uparrow_J X) \cup (\uparrow_J Y) = \uparrow_J (X \cup Y)$.*

Proof. Let $x \in (\uparrow_J X) \cup (\uparrow_J Y)$. If $x \in \uparrow_J X$, there is $x' \in X$ such that $\pi_J(x') = \pi_J(x)$. We also have $x' \in X \cup Y$, which proves $x \in \uparrow_J (X \cup Y)$. The case where $x \in \uparrow_J Y$ is similar.

In the other direction, let $x \in \uparrow_J (X \cup Y)$, there exists $x' \in X \cup Y$ such that $\pi_J(x) = \pi_J(x')$. If $x' \in X$ then $x \in \uparrow_J X$ and otherwise $x \in \uparrow_J Y$. Therefore $x \in (\uparrow_J X) \cup (\uparrow_J Y)$.

We use the following to restrict our study to bounded polygone, so that we can easily represent them by there extremal points.

Lemma 14.

$$\text{value}(s) = \downarrow ([-W, W]^d \cap \text{value}(s))$$

Proof. Let $v \in \text{value}(s)$, there is a strategy σ_{\exists} whose outcome all have weight greater than v . Since all outcome ρ of the game on all coordinate i , a weight $w_i(\rho)$ greater than $-W$, we can replace all v_i that are smaller than $-W$ by $-W$ and we obtain a $v' \geq v$ such that $v' \in \text{value}(s)$. Similarly, all weights are smaller than W so for all coordinate i , v_i cannot be greater than W . Therefore $v' \in [-W, W]^d \cap \text{value}(s)$ and $v \downarrow ([-W, W]^d \cap \text{value}(s))$.

For the other inclusion, this is simply because $\text{value}(s)$ is downward closed and $[-W, W]^d \cap \text{value}(s) \subseteq \text{value}(s)$.

We will also use the following lemma, to apply the \uparrow operator only on bounded polytopes.

Lemma 15. *Given a set $X \subseteq \mathbb{R}^d$, a bound W and $J \subseteq \llbracket 1, d \rrbracket$, we have that:*

$$[-W, W]^d \cap (\uparrow_J [-W, W]^d \cap (\downarrow [-W, W]^d \cap X)) = [-W, W]^d \cap (\uparrow_J \downarrow ([-W, W]^d \cap X))$$

Proof. The inclusion holds because $[-W, W]^d \cap (\downarrow [-W, W]^d \cap X) \subseteq \downarrow [-W, W]^d \cap X$ and \uparrow_J is monotonic with respect to inclusion.

In the other direction, let $x \in [-W, W]^d \cap (\uparrow_J \downarrow ([-W, W]^d \cap X))$. There is $x' \in \downarrow ([-W, W]^d \cap X)$ such that $\pi_J(x') = \pi_J(x)$. There is $x'' \in [-W, W]^d \cap X$ such that $x'' \geq x'$. Let y be such that $y_j = x''_j$ if $j \in J$ and $y_j = x_j = x'_j \leq x''_j$

otherwise. We have $\pi_J(y) = x$ and $y \in [-W, W]^d$ because both x'' and x do. We also have $y \in \downarrow [-W, W]^d \cap X$ because it is smaller than x'' . This shows that $x \in \uparrow_J \{y\} \subseteq [-W, W]^d \cap (\uparrow_J [-W, W]^d \cap (\downarrow [-W, W]^d \cap X))$. \square

Theorem 14. *There is polynomial function P_6 such that for all weighted game \mathcal{G} and state s of \mathcal{G} , $\text{value}(s)$ is of the form $\bigcup_{g \in G} C_g$ where g is finite and for each g , C_g is a convex set, which can be given as the intersection of a finite set of half-spaces $H_1^g, \dots, H_{k_g}^g$ whose associated inequations have size bounded by $P_6(\|W\|, \|n\|, d)$.*

Proof. By Thm. 13 $\text{value}(s)$ is equal to:

$$\bigcup_{\{s\} \subseteq S \subseteq \text{Stat}} \bigcap_{s' \in S} \bigcap_{j \in J} \uparrow_{J \setminus \{j\}} \bigcap_{\sigma_V \in \mathbb{M}} \bigcup_{S' \in \text{SCC}(s', \sigma_V)} \downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in \mathbb{C}(S') \right\} \right)$$

We can then reorder intersection and union operators using the fact that \uparrow operators are compatible with union (Lem. 13). We do this by considering functions that selects for a tuple (s', j, σ_V) a strongly connected component of $\text{SCC}(s', \sigma_V)$, i.e. functions of F , where $F = \{f: \text{Stat} \times J \times \mathbb{M} \mapsto 2^{\text{Stat}} \mid \forall s' \in \text{Stat}, j \in J, \sigma_V \in \mathbb{M}. f(s', j, \sigma_V) \in \text{SCC}(s', \sigma_V)\}$. Note that the set F can be huge, but is always finite. We then have that $\text{value}(s)$ is equal to:

$$\bigcup_{\{s\} \subseteq S \subseteq \text{Stat}} \bigcup_{f \in F} \bigcap_{s' \in S} \bigcap_{j \in J} \uparrow_{J \setminus \{j\}} \bigcap_{\sigma_V \in \mathbb{M}} \downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in f(s', j, \sigma_V) \right\} \right).$$

We use Lem. 14 to restrict the difficult operations (\downarrow and \uparrow) to bounded polytopes. This is because their results are easier to express using the vertice representation, which requires bounded polytopes. We express $\text{value}(s)$ as:

$$\downarrow [-W, W]^d \cap \bigcup_{\{s\} \subseteq S \subseteq \text{Stat}} \bigcup_{f \in F} \bigcap_{s' \in S} \bigcap_{j \in J} \uparrow_{J \setminus \{j\}} \bigcap_{\sigma_V \in \mathbb{M}} \downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in f(s', j, \sigma_V) \right\} \right).$$

Note that we can also use Lem. 14 to show:

$$\downarrow [-W, W]^d \cap \bigcap_{\sigma_V \in \mathbb{M}} \downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in f(s', j, \sigma_V) \right\} \right) = \bigcap_{\sigma_V \in \mathbb{M}} \downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in f(s', j, \sigma_V) \right\} \right)$$

since this set is the value set for the game where J would be empty (see 8). We therefore write $\text{value}(s)$ as:

$$\downarrow [-W, W]^d \cap \bigcup_{\{s\} \subseteq S \subseteq \text{Stat}} \bigcup_{f \in F} \bigcap_{s' \in S} \bigcap_{j \in J} \uparrow_{J \setminus \{j\}} \downarrow [-W, W]^d \cap \bigcap_{\sigma_V \in \mathbb{M}} \downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in f(s', j, \sigma_V) \right\} \right).$$

We now use Lem. 15 to introduce a cube $[-W, W]^d$ between the \uparrow and \downarrow operators. Then $\text{value}(s)$ can be written as:

$$\bigcup_{\{s\} \subseteq S \subseteq \text{Stat}} \bigcup_{f \in F} \downarrow [-W, W]^d \cap \bigcap_{s' \in S} \bigcap_{j \in J} \uparrow_{J \setminus \{j\}} [-W, W]^d \cap \bigcap_{\sigma_V \in \mathbb{M}} \downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in f(s', j, \sigma_V) \right\} \right).$$

We obtain therefore a finite union of convex intersections. We now use geometrical arguments to show that the inequations defining these polyhedra are of polynomial size which will conclude the proof.

Thanks to Thm. 3.6, the set $\bigcap_{\sigma_V \in \mathbb{M}} \downarrow \text{conv} \left(\left\{ \frac{1}{|c|} \cdot w(c) \mid c \in f(s', j, \sigma_V) \right\} \right)$ can be given by the intersection of half-spaces whose inequations have size bounded by $P_2(d \cdot (\|W\| + \|n+1\| + 1) + \|n+1\|, d)$. The intersection with the cube $[-W, W]^d$ is obtained by adding inequations of size $d + \|W\|$ which is within the bounds of P_2 . To apply \downarrow we go back to the point representation, using Thm. 3.5, sizes of the vertices are bounded by $P_3(P_2(d \cdot (\|W\| + \|n+1\| + 1) + \|n+1\|, d), d)$. Then using Thm. 3.6 once again, the downward closure gives us half-spaces which have size bounded by $P_2(P_3(P_2(d \cdot (\|W\| + \|n+1\| + 1) + \|n+1\|, d), d), d)$. Taking the intersection with the cube $[-W, W]^d$ again adds inequation that are with the bounds of the polynomial function. We go back to a point representation in order to apply $\uparrow_{J \setminus \{j\}}$, which gives, thanks to Thm. 3.5, points of size bounded by $P_3(P_2(P_3(P_2(d \cdot (\|W\| + \|n+1\| + 1) + \|n+1\|, d), d), d), d)$.

We show that the operator $\uparrow_{J \setminus \{j\}}$ gives half-spaces whose size is polynomially bounded. We have seen in Lem. 3 that given a polyhedron P :

$$\uparrow_J P = \bigcap_{H \in \mathcal{F}(\pi_J(P))} H + B_J$$

When P is given by its vertices, the projection π_J is obtained by projecting points one by one, which means replacing coordinates of J by 0 and therefore does not increase the size of the points. The equations to describe half-spaces of $\mathcal{F}(\pi_J(P))$, can be of size bounded by P_2 with respects to the vertices thanks to Thm. 3.4. The same system of inequations, gives the sets $H + \mathbb{R}^J$ when lifted to \mathbb{R}^d . Therefore the size the inequations we obtain after applying $\uparrow_{J \setminus \{j\}}$ is bounded by $P_6(\|W\|, \|n\|, d) = P_2(P_3(P_2(P_3(P_2(d \cdot (\|W\| + \|n+1\| + 1) + \|n+1\|, d), d), d), d), d)$. Then the intersection with the cube $[-W, W]^d$ does not add any complexity and thus the polynomial function P_6 bounds the size of the half-spaces that describes the convex set of the expression of $\text{value}(s)$. \square

Algorithm We deduce a $\Sigma_2\text{P}$ algorithm from the previous theorem.

Theorem 15. *There is a polynomial function P such that if S is a set of states and λ a system of linear equations, then $\bigcap_{s \in S} \text{value}(s) \cap [-W, W]^d \cap \text{polyhedron}(\lambda)$ is of the form $\bigcup_{f \in F} \text{conv}(E_f)$ where F is finite and for each $f \in F$, E_f is a set of vertices which all have size bounded by $P(|\text{Stat}|, d, \|\lambda\|, \|W\|)$.*

Proof. By Thm. 14 we have a finite union of polyhedra whose equations have size bounded by a polynomial function. Let $\lambda = (a_1, b_1) \dots (a_k, b_k)$. We use Thm. 7 to say that if $\text{polyhedron}(\lambda)$ intersects this union of polyhedra, then there is a solution of size bounded by $P_4(\max\{\|a_i, b_i\| \mid 1 \leq i \leq k\}, P_6(\|W\|, \|n\|, d), d)$.

Theorem 16. *The polyhedron value problem is in $\Sigma_2\text{P}$.*

The algorithm relies on the same principle than for the lim inf but uses Theorem 14 to bound the size of the guessed point. Correctness follows from Thm. 15.

Algorithm 1: Algorithm for the polyhedron value problem in multidimensional mean-payoff games

Existentially guess v in X
 | if $x \in \bigcap_{s \in S} \text{value}(s) \cap \text{polyhedron}(\lambda)$ then return true;
 | else return false;

6 Approximate Pareto Curve

In this section, we show how to approximate the Pareto curve in the sense of [10]. We consider both a *relative* and an *absolute* notions of approximation.

Definition 3. An ε -relative approximate Pareto curve [10], is a set P_ε of thresholds $v \in \mathbb{Q}^d$ that Eve can enforce such that there is no threshold v' that Eve can force such that $v' \geq (1 + \varepsilon)v$. An ε -absolute-approximate Pareto curve, is a set P_ε of thresholds v that Eve can enforce such that there is no threshold v' that Eve can force such that $v' \geq v + (\varepsilon, \dots, \varepsilon)$.

The relative notion of approximation can be used when all the weights are strictly positive only (it is not well defined for negative vectors, and not an approximation in 0). For the general case, we consider the absolute (or additive) version of the approximation.

We show that if we fix the number of dimensions d , then there is an pseudo-polynomial⁵ time algorithm to construct an approximate Pareto curve for the two notions of approximation.

Theorem 17. Let d be a fixed number of dimension, let W be the largest absolute value of the weights in the game. There is a polynomial time algorithm (in the size of the game and W), that given a weighted game \mathcal{G} , constructs both an ε -relative approximate and an ε -absolute approximate Pareto curve.

Proof. For the absolute Pareto curve, the justification is as follows. The algorithm works by considering a grid on $[-W, W]^d$ and checking for each point on this grid whether Eve can ensure the corresponding threshold. We consider the set G of points x inside $[-W, W]^d$ which have coordinates of the form $x_i = \varepsilon \cdot k$ with $k \in \mathbb{Z}$. The number of points on the grid G is polynomial thanks to the hypothesis that W is encoded in unary and that the number of dimensions d is fixed. We can determine for each point whether there is a strategy of Eve to ensure the corresponding threshold. Thanks to [4, Thm. 1], to determine that, there is an algorithm that works in time $O(n^2 \cdot m \cdot d \cdot W \cdot (d \cdot n \cdot W)^{d^2+2 \cdot d+1})$ (where n is the number of states, and m the number of edges), which is polynomial here. Therefore we can compute in polynomial time a set P_ε of points x in G such that x can be ensured by Eve and there is no $x' \in G \setminus \{x\}$ that can be ensured and such that $x' \geq x$.

⁵ That is a polynomial time algorithm for weights encoded in unary.

We show that P_ε is an ε -absolute-approximate Pareto curve. Let $s \in P_\varepsilon$, and $s' \geq s + (\varepsilon, \dots, \varepsilon)$. If $s + (\varepsilon, \dots, \varepsilon) > (W, \dots, W)$ then s' is above the maximum possible weights and cannot be ensured. Otherwise there is $s + (\varepsilon, \dots, \varepsilon) \in G$. Since $s \in P_\varepsilon$ and by construction of P_ε , $s + (\varepsilon, \dots, \varepsilon)$ cannot be ensured by **Eve**. Since s' is greater than $s + (\varepsilon, \dots, \varepsilon)$, it cannot be ensured either. This shows that P_ε is an ε -absolute-approximate Pareto curve.

For the relative Pareto curve, the justification is as follows. Here, we need only to consider games with strictly positive weights. The minimum weight that can be obtained in the game is $(1, \dots, 1)$ and the maximum is (W, \dots, W) . Since the weights are all greater than 1, the ε -absolute-approximate Pareto curve is also a ε -approximate Pareto curve and therefore the preceding algorithm is suitable to construct a ε -approximate Pareto curve.

The two results rely on the following properties. First, when the number of dimension is fixed then for both notions of approximation, only a pseudo-polynomial number of thresholds are relevant for the approximation of the Pareto curve. For the absolute approximation, there are the minimal points of the hyper-rectangles defined by the grid of precision ε for all value between $-W$ and W , for the relative approximation, there are the minimal points of hyper-rectangles, such that, in each dimension, the ratio of the larger to the smaller coordinate is $1 + \varepsilon$. Second, when the number of dimensions is fixed and weights are encoded in unary, [4] gives a polynomial time algorithm to decide the threshold problem. So the overall complexity of our algorithm is pseudo-polynomial.

In the case of weights encoded in binary, we show that even with one dimension, computing an ε -absolute-approximate Pareto curve is harder than computing the winner of a mean-payoff game (a similar result of inapproximability has been proved in [5]). Thus we cannot approximate the Pareto curve in polynomial time unless there is a polynomial time algorithm to compute the winner of a mean-payoff game which is an long standing open problem in the area [14].

Theorem 18. *If there is a polynomial algorithm that given a weighted game \mathcal{G} with dimension $d = 1$, construct a ε -absolute-approximate Pareto curve, then there is a polynomial algorithm that given a weighted game \mathcal{G} with dimension $d = 1$ decide whether **Eve** can ensure the threshold 0.*

Proof. Assume that a polynomial algorithm exist for constructing a ε -absolute-approximate Pareto curve. We know (see for instance Thm. 8) that in a one dimensional mean-payoff game, the value (i.e. maximal threshold that can be ensured) is a rational whose denominator is bounded by the number of vertices n . Let us choose $\varepsilon = \frac{1}{2n}$ (which is polynomial) and construct a ε -absolute-approximate Pareto curve P_ε in polynomial time. We can look at the points of P_ε to determine if one of them is above $-\varepsilon$. This is the case if and only if there is a winning strategy in the mean-payoff game for the threshold 0. Thus we have constructed a polynomial algorithm that decides whether **Eve** can ensure the threshold 0.

References

1. R. Alur, A. Degorre, O. Maler, and G. Weiss. On omega-languages defined by mean-payoff conditions. In *FoSSaCS'09*, pages 333–347, 2009.
2. P. Bouyer, R. Brenguier, N. Markey, and M. Ummels. Concurrent games with ordered objectives. In L. Birkedal, editor, *FoSSaCS'12*, volume 7213 of *Lecture Notes in Computer Science*, pages 301–315. Springer-Verlag, Mar. 2012.
3. A. Chakrabarti, L. de Alfaro, T. A. Henzinger, and M. Stoelinga. Resource interfaces. In *EMSOFT 2003*, volume 2855 of *Lecture Notes in Computer Science*. Springer, 2003.
4. K. Chatterjee and Y. Velner. Hyperplane separation technique for multidimensional mean-payoff games. In *CONCUR 2013–Concurrency Theory*, pages 500–515. Springer, 2013.
5. R. Gentilini. A note on the approximation of mean-payoff games. *Inf. Process. Lett.*, 114(7):382–386, 2014.
6. M. Joswig. Beneath-and-beyond revisited. In *Algebra, Geometry and Software Systems*, pages 1–21. Springer, 2003.
7. M. Jurdziński. Deciding the winner in parity games is in UP and in co-UP. *Information Processing Letters*, 68(3):119–124, 1998.
8. D. A. Martin. Borel determinacy. *The annals of Mathematics*, 102(2):363–371, 1975.
9. J. Matoušek. *Lectures on discrete geometry*, volume 212. Springer, 2002.
10. C. H. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 86–92. IEEE, 2000.
11. A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *POPL89*, pages 179–190. ACM Press, 1989.
12. A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.
13. Y. Velner, K. Chatterjee, L. Doyen, T. A. Henzinger, A. Rabinovich, and J.-F. Raskin. The complexity of multi-mean-payoff and multi-energy games. *CoRR*, abs/1209.3234, 2012.
14. U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *TCS*, 158(1):343–359, 1996.