


Hitting and Covering Affine Families of Convex Polyhedra, with Applications to Robust Optimization

Jean Cardinal ✉ 

Université libre de Bruxelles (ULB), Brussels, Belgium

Xavier Goaoc ✉ 

Université de Lorraine, CNRS, INRIA, LORIA, Nancy, F-54000, France

Sarah Wajsbrodt ✉ 

Université de Lorraine, CNRS, INRIA, LORIA, Nancy, F-54000, France

Abstract

Geometric hitting set problems, in which we seek a smallest set of points that collectively hit a given set of ranges, are ubiquitous in computational geometry. Most often, the set is discrete and is given explicitly. We propose new variants of these problems, dealing with continuous families of convex polyhedra, and show that they capture decision versions of the two-level finite adaptability problem in robust optimization. We show that these problems can be solved in strongly polynomial time when the size of the hitting/covering set and the dimension of the polyhedra and the parameter space are constant. We also show that the hitting set problem can be solved in strongly quadratic time for one-parameter families of convex polyhedra in constant dimension. This leads to new tractability results for finite adaptability that are the first ones with so-called left-hand-side uncertainty, where the underlying problem is non-linear.

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Problems, reductions and completeness

Keywords and phrases Geometric hitting set problem, Continuous families of polyhedra, Robust optimization

Digital Object Identifier 10.4230/LIPIcs.MFCS.2025.33

Related Version *Full Version*: <https://arxiv.org/abs/2504.16642> [12]

Acknowledgements The authors thank Safia Kedad-Sidhoum, Anton Medvedev and Frédéric Meunier for helpful discussions on finite adaptability and Guillaume Moroz for helpful discussions on computer algebra.

1 Introduction

In this paper we present three contributions on hitting and covering problems for families of convex polyhedra. (Throughout the paper, all polyhedra considered are convex.) First, we introduce a new flavor of these problems, dealing with hitting or covering *continuous* families of polyhedra, and show that they capture decision versions of the two-level finite adaptability problem in robust optimization. Next, we show that general methods from quantifier elimination yield strongly polynomial time algorithms for these problems when certain parameters are constant. Last, we present a fixed-parameter algorithm for a special case of our continuous hitting set problem.



© Jean Cardinal, Xavier Goaoc, and Sarah Wajsbrodt;
licensed under Creative Commons License CC-BY 4.0

50th International Symposium on Mathematical Foundations of Computer Science (MFCS 2025).

Editors: Paweł Gawrychowski, Filip Mazowiecki, and Michał Skrzypczak; Article No. 33; pp. 33:1–33:18

Leibniz International Proceedings in Informatics



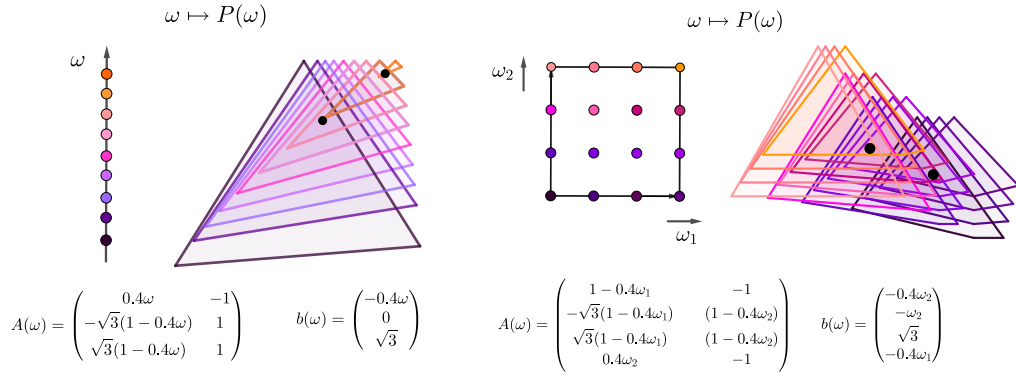
LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1.1 Hitting affine families of polyhedra

An *affinely parameterized family of polyhedra* in \mathbb{R}^d , or an *affine family of polyhedra* for short, is a continuous family $P(\Omega)$ of polyhedra in \mathbb{R}^d defined by a domain $\Omega \subset \mathbb{R}^p$ and two affine maps $A : \Omega \mapsto \mathbb{R}^{m \times d}$ and $b : \Omega \mapsto \mathbb{R}^m$ via

$$P(\Omega) = \{P(\omega) : \omega \in \Omega\} \quad \text{where} \quad P(\omega) = \{x \in \mathbb{R}^d : A(\omega)x \leq b(\omega)\}. \quad (1)$$

Observe that an affine family of polyhedra has three defining parameters: the dimension d of the ambient space of the polyhedra, the dimension p of the parameter space Ω , and the number m of constraints defining each polyhedron. When every polyhedron $P(\omega)$ is bounded, we call $P(\Omega)$ an *affine family of polytopes*.



■ **Figure 1** Two examples of affine families of polygons together with a hitting set (the black dots); For readability, we represent the polygon $P(\omega)$ for only a finite subset of ω in the domain.

We call a set $S \subset \mathbb{R}^d$ a *hitting set* for an affine family of polyhedra if S intersects every member of that family. See Figure 1 for some examples. We adopt the following important convention: if an affine family of polyhedra $P(\Omega)$ has one empty member, that is if $P(\omega) = \emptyset$ for some $\omega \in \Omega$, then $P(\Omega)$ has no hitting set. We first consider the following computational problem:

Given an integer k and an affine family of polyhedra, does that family admit a hitting set of size k ?

1.2 Context: robust optimization and finite adaptability

This continuous hitting set problem arises as the decision version of a special case of the *finite adaptability* problem in robust optimization [6]. This line of research in mathematical programming deals with uncertainty in planning by modeling the decision to make as the optimization of some objective function under some constraints, with the objective function and the constraints depending not only of the free variables, but also of some uncertainty parameter ω . One then searches for an optimal solution that is valid for all values of the uncertainty parameter ω . Among robust optimization problems, those that involve successive stages of decision are of particular interest [39]; Here is for instance the *two-stage robust*

optimization problem:

$$\begin{aligned} \inf_{x_f \in \mathbb{R}^\ell} \quad & \sup_{\omega \in \Omega} \quad \inf_{x_s \in \mathbb{R}^\delta} \quad c_f^T x_f + c_s(\omega)^T x_s \\ \text{s. t.} \quad & A_f(\omega)x_f + A_s(\omega)x_s \leq b(\omega) \end{aligned} \quad (2)$$

where Ω is the domain of uncertainty, $A_f(\omega)$, $A_s(\omega)$, $b(\omega)$ and $c_s(\omega)$ are input matrices and vectors depending on the uncertainty parameter ω , and c_f is a deterministic input vector. The variables x_f and x_s correspond respectively to the first and second stages of the optimization, where the second stage takes place only once the uncertainty ω has been revealed. The problem is therefore to optimize in the first stage the worst-case outcome from the second stage.

In 2010, Bertsimas and Caramanis [8] proposed to approximate the solution of Problem (2) by solving the following *finite adaptability* problem:

$$\begin{aligned} \inf_{\substack{x_f \in \mathbb{R}^\ell \\ x_s^1, x_s^2, \dots, x_s^k \in \mathbb{R}^\delta}} \quad & \sup_{\omega \in \Omega} \quad \inf_{i \in [k]} \quad c_f^T x_f + c_s(\omega)^T x_s^i \\ \text{s. t.} \quad & A_f(\omega)x_f + A_s(\omega)x_s^i \leq b(\omega) \end{aligned} \quad (3)$$

(We use $[k]$ to denote the set $\{1, 2, \dots, k\}$.) For fixed k this problem is called the *k-adaptability problem*. It models the precomputation, in the first stage, of k candidate values for the second-stage variable x_s . Once the uncertainty ω is revealed, the second stage consists in selecting one of the k precomputed values that satisfies the constraints and minimizes the objective. Under a suitable continuity assumption, the value of Problem (3) converges to the value of (2) as $k \rightarrow \infty$ [27, § 2]. See [24, 33, 11, 36] for recent work on Problem (3) and variants.

Throughout the paper we only consider the case where A_f , A_s , b , and c_s are affine maps.

1.3 Our results

Problem correspondence. The continuous hitting set problem stated above corresponds to the decision version of the optimization Problem (3) when there is no first-stage decision ($\ell = 0$); See Lemma 5. This correspondence is also the motivation for our convention that an affine family of polyhedra with one empty member has no hitting set, as such a family corresponds to an unfeasible problem. Like in the discrete setting, the hitting and covering problems for affine families of polyhedra enjoy some form of duality (see Section 2.1). We prove (Lemma 6) that the decision version of the general finite adaptability problem is a special case of the following covering problem:

Given two affine families of polyhedra in \mathbb{R}^d and $k \in \mathbb{N}$, does there exist k polyhedra in the second family whose union covers a polyhedron from the first family?

Notice that this problem is purely linear, whereas the finite adaptability Problem (3) exhibits some non-linearity.

Complexity. We then turn our attention to the computational complexity of these problems. We first observe that quantifier elimination methods yield strongly polynomial solutions¹ for these problems in fixed dimension.

¹ The standard model of computation in computational geometry is the *Real-RAM* model, a variant of the classical random access machine (RAM) that operates on arbitrary real numbers instead of finite

► **Theorem 1.** *For every constant k, ℓ, δ, p , there is an algorithm that solves the decision version of the k -adaptability Problem (3) on m constraints, with Ω a polyhedron in \mathbb{R}^p given as an intersection of v halfspaces, in time strongly polynomial in m and v .*

Similarly, the hitting and covering problems stated above can be solved in time $m^{O(1)}$ when we fix k , the ambient dimensions of the polyhedra and the dimension of the parameter space(s) (see Section 3). The bound in the exponent grows with the values of the fixed parameters, and the fixed-parameter tractability of these problems is a natural question. In the special case where the parameter domain is one-dimensional, we can give an algorithm for the hitting set problem where the order of the complexity does not depend on k or the ambient dimension of the polyhedra.

► **Theorem 2.** *For every constant d and k , there is an algorithm that decides in strongly $O(m^2)$ time, given an affine family of polytopes $P([\alpha, \beta])$ in \mathbb{R}^d , each defined by at most m constraints, whether $P([\alpha, \beta])$ admits a hitting set of size k .*

If k is part of the input, then our algorithm has complexity $O(km^2)$ in the Real-RAM model (Proposition 13) but is no longer strongly polynomial. We can extend the method behind Theorem 2 to the optimization problem:

► **Corollary 3.** *For every constant δ and k , there is an algorithm that solves the k -adaptability Problem (3) in the special case where $\ell = 0$ and Ω is an interval, in time strongly $O(m^4)$.*

1.4 Background and related works

1.4.1 Complexity of finite adaptability

Bertsimas and Caramanis proved [8, Prop. 5] that unless $P = NP$, there is no polynomial-time algorithm for the finite adaptability problem (3). This already holds in the special case where $k = 2$, there is no first decision x_f , and A_s is constant, *i.e.* independent of ω . Their proof requires, however, that the dimensions of x_s and ω as well as the number m of rows in A_s be part of the input. A natural question is whether the computational complexity becomes polynomial when some of these parameters are constant.

We are only aware of two² previous tractability results for Problem (3). On the one hand, Subramanyam, Gounaris and Wiesemann [36, Prop. B.3] observed that Problem (3) reduces to linear programming, and is therefore polynomial, when A_f , A_s and b are constant (but c_s still depends on ω). On the other hand, Kedad-Sidhoum, Meunier and Medvedev [27, Theorems 1.1 and 1.2] proved that if $k \leq 3$, the number of vertices (and dimension) of Ω is bounded, and only the right-hand side b of the constraints depends on the uncertainty parameter (that is, A_f , A_s and d are constant), then Problem (3) reduces to a constant number of linear programs and is therefore polynomial in the Turing machine model.

The tractable cases of the decision version of finite adaptability that are identified by Theorem 1 are therefore new in that they allow *left-hand-side uncertainty*, that is with the map A_s affine rather than constant. This makes the underlying problem non-linear in x_f, x_s

binary words, see [20, §7.4] and [21, §6.1]. Algorithms that run in polynomial time on a Real-RAM and that translate to polynomial-time algorithms in the usual sense are called *strongly polynomial*. Whether linear programming can be solved in strongly polynomial time remains a major open problem, known as Smale's 9th problem [34].

² Note that Bertsimas and Caramanis gave a reformulation of Problem (3) as a bilinear optimization problem with a linear objective function [8, Prop. 4], but it does not yield a polynomial-time algorithm even when all dimensions and k are fixed. Indeed, their reformulation uses $\ell + d + m^k$ variables and $m^k(v + 2)$ bilinear constraints, where v is the number of extreme points of the domain Ω .

and ω . An example of problem from operations research that directly benefits from this tractability result is *antenna design* [7, § 2]. Indeed, it can be modeled by two-stage robust optimization (2) with continuous variables and uncertainty, and the recourse matrices A_f and A_s depend on the uncertainty parameter. Moreover, the fact that the recourse matrices are uncertain makes affine decision rules, the standard approximation method for TSRO [5, § 14.3.1], potentially intractable [5, § 14.3.3]. Other examples include variants of project management [5, example 14.2.1], supply chain design [23, § 4.1], route planning [23, § 4.2] and capital budgeting [23, § 4.3]. The variants are obtained in two steps: relaxing the discrete variables into continuous ones and making the model more expressive by allowing the recourse matrices to depend on the uncertainty. Both steps are common practice in operations research, so these variants are of interest.

Corollary 3, which shows that without first decision and with one-dimensional uncertainty, finite adaptability can be solved in $O(m^4)$ time for any constant k and δ , is the first fixed-parameter tractability result for this problem that we are aware of.

1.4.2 Related notions in computational geometry

Hitting set problems were extensively investigated in discrete and computational geometry for discrete, unstructured families of sets. One typically expects that deciding whether k points suffice to hit n given subsets of \mathbb{R}^d is NP-hard when k or d is part of the input; see for instance the landmark results of Megiddo [31] and Megiddo and Supowit [32]. When both k and d are fixed, the problem can be solved in polynomial time via the computation of an arrangement.

Notions of uncertainty related to robust mathematical programming were also investigated by computational geometers. In this context, uncertainty in the input geometric data is typically modeled by replacing single points by regions (say, disks of some small radius), in which the actual point is guaranteed to lie. One can preprocess the regions, and prepare for the worst-case actual input. Recent examples include sorting and finding the Pareto front of imprecise point sets [38, 37]. We refer to Löffler for a general discussion of *imprecision in computational geometry* [28].

More generally, continuous families of geometric sets are ubiquitous in computational geometry. In *range searching*, for instance, we are given a collection of points in d -dimensional space, and are asked to construct a data structure that allows to answer queries of the following form: Given a *range* $R \subset \mathbb{R}^d$, report the set of points from the collection that are contained in R . The ranges are typically restricted to be axis-aligned boxes, halfspaces, or semialgebraic sets, giving rise to the corresponding *orthogonal*, *halfspace*, or *semialgebraic* range searching problems; see the survey of Agarwal [1] for references. In the problem of finding small ε -net, the goal is similar to ours, in that we wish to find small hitting sets of continuous family of ranges. More precisely, given a finite set S of points together with a continuous family of ranges (typical examples are all halfspaces, or all unit disks), we wish to find a set N of points that collectively hit all ranges that contain at least an ε fraction of the points of S [25]. If the set is restricted to be a subset of S , then N is called a *strong ε -net*, and a *weak ε -net* otherwise [15]. While this also bears a superficial resemblance with our question, it is yet quite different, since in our case, all members of the input continuous family have to be hit, whatever their size is. Brönnimann and Goodrich [10], and then Even, Rawitz, and Shahar [22], showed that finding ε -nets of size $O(c/\varepsilon)$ in polynomial time implied the existence of c -approximation algorithms for the discrete hitting set problem. In the same spirit, Elbassioni [19] recently proposed an approximation algorithm for finding small hitting sets of infinite range spaces of bounded VC-dimension, and gave an application to the problem of covering a polygonal region by translates of a given polygon.

2 Relation between hitting/covering and finite adaptability

Let us clarify the relation between the hitting and covering problems for affine families of polytopes and the finite adaptability Problem (3). We assume in this section that the domain of uncertainty Ω is a polyhedron in \mathbb{R}^p , the matrices $A_f(\omega)$, $A_s(\omega)$ and the vectors $b(\omega)$, $c_s(\omega)$ depend affinely on the uncertainty parameter ω , and the vector c_f is deterministic.

2.1 The hitting-covering duality

Let us start by remarking that for any regions $\Lambda \subseteq \Omega \subseteq \mathbb{R}^p$, every affine map on Ω restricts, in a unique way, to an affine map on Λ ; Conversely, every affine map on Λ extends to an affine map on Ω , and this extension is unique if Ω is contained in the affine span of Λ . This implies that any affine family of polyhedra $P(\Omega)$ restricts (in a unique way) to an affine family of polyhedra $P(\Lambda)$, and that conversely every affine family of polyhedra $P(\Lambda)$ extends to an affine family of polyhedra $P(\Omega)$, and that this extension is unique if Ω is contained in the affine span of Λ .

Let us fix two affine maps $A : \mathbb{R}^p \mapsto \mathbb{R}^{m \times d}$ and $b : \mathbb{R}^p \mapsto \mathbb{R}^m$ and consider the maps

$$\begin{aligned} P : \omega &\mapsto \{x \in \mathbb{R}^d : A(\omega)x \leq b(\omega)\} \quad \text{for } \omega \in \mathbb{R}^p, \\ \hat{P} : x &\mapsto \{\omega \in \mathbb{R}^p : A(\omega)x \leq b(\omega)\} \quad \text{for } x \in \mathbb{R}^d, \end{aligned}$$

so that $x \in P(\omega)$ if and only if $\omega \in \hat{P}(x)$. Hence, switching from P to \hat{P} exchanges hitting and covering in the sense that for any domain $\Omega \subseteq \mathbb{R}^p$, a set $S \subseteq \mathbb{R}^d$ is a hitting set for $\{P(\omega) : \omega \in \Omega\}$ if and only if $\Omega \subseteq \bigcup_{x \in S} \hat{P}(x)$. It turns out that \hat{P} is also an affine family of polyhedra, which we call the *dual* of P .

► **Lemma 4.** *If $P(\mathbb{R}^p)$ is an affine family of polyhedra in \mathbb{R}^d , each defined by m constraints, then $\hat{P}(\mathbb{R}^d)$ is an affine family of polyhedra in \mathbb{R}^p , each defined by m constraints.*

Proof. Let us fix a basis (e_1, e_2, \dots, e_p) of \mathbb{R}^p , so as to write $\omega = (\omega_1, \omega_2, \dots, \omega_p)$, $A(\omega) = A_0 + \sum_{i=1}^p \omega_i A_i$ and $b(\omega) = b_0 + \sum_{i=1}^p \omega_i b_i$, where $A_0, A_1, \dots, A_p \in \mathbb{R}^{m \times d}$, $b_0, b_1, \dots, b_p \in \mathbb{R}^m$. For $x \in \mathbb{R}^d$, we have

$$\begin{aligned} \hat{P}(x) &= \left\{ \omega \in \mathbb{R}^p : \left(A_0 + \sum_{i=1}^p \omega_i A_i \right) x \leq b_0 + \sum_{i=1}^p \omega_i b_i \right\} \\ &= \left\{ \omega \in \mathbb{R}^p : \sum_{i=1}^p (A_i x - b_i) \omega_i \leq b_0 - A_0 x \right\}. \end{aligned}$$

Let $\hat{A}(x) = \sum_{i=1}^p (A_i x - b_i) e_i^T \in \mathbb{R}^{m \times p}$, so that $\hat{A}(x)\omega = \sum_{i=1}^p (A_i x - b_i) \omega_i$. Let $\hat{b}(x) = b_0 - A_0 x$. Note that $\hat{P}(x) = \{\omega \in \mathbb{R}^p : \hat{A}(x)\omega \leq \hat{b}(x)\}$ with \hat{A} and \hat{b} affine maps. It follows that $\hat{P}(\mathbb{R}^d)$ is an affine family of polyhedra, with ambient space of dimension p , parameter space of dimension d and each polyhedron defined by m constraints. ◀

When Ω is a polyhedron in \mathbb{R}^p , a natural candidate for the dual of an affine family $P(\Omega)$ of polyhedra is $\{\omega \in \Omega : x \in P(\omega)\}$. This is again an affine family of polyhedra, which we denote by $\hat{P}_\Omega(\mathbb{R}^d) = \{\hat{P}(x) \cap \Omega : x \in \mathbb{R}^d\}$. In other words, restricting $P(\mathbb{R}^p)$ into $P(\Omega)$ translates, under duality, into intersecting each dual polyhedron with Ω .

2.2 Direct correspondences

Let us first consider the k -adaptability problem (3) without first decision ($\ell = 0$), that is

$$\begin{aligned} \inf_{x_s^1, x_s^2, \dots, x_s^k \in \mathbb{R}^d} \quad & \sup_{\omega \in \Omega} \quad \inf_{i \in [k]} \quad c_s(\omega)^T x_s^i \\ \text{s. t.} \quad & A_s(\omega) x_s^i \leq b(\omega) \end{aligned} \quad (4)$$

► **Lemma 5.** *For any real t , the value of the k -adaptability problem without first decision (4) is at most t if and only if the affine family of polyhedra $P_t(\Omega) = \{P_t(\omega) : \omega \in \Omega\}$ defined by*

$$P_t(\omega) = \left\{ x \in \mathbb{R}^d : \begin{pmatrix} A_s(\omega) \\ c_s(\omega)^T \end{pmatrix} x \leq \begin{pmatrix} b(\omega) \\ t \end{pmatrix} \right\}$$

admits a hitting set of size k .

Proof. Let $t \in \mathbb{R}$. The value of Problem (4) is at most t if and only if there exists $x_s = (x_s^1, x_s^2, \dots, x_s^k) \in (\mathbb{R}^d)^k$ such that

$$\begin{aligned} & \sup_{\omega \in \Omega} \quad \inf_{i \in [k] \text{ s.t. } A_s(\omega) x_s^i \leq b(\omega)} c_s(\omega)^T x_s^i(\omega) \leq t, \\ \Leftrightarrow \quad & \forall \omega \in \Omega, \quad \inf_{i \in [k] \text{ s.t. } A_s(\omega) x_s^i \leq b(\omega)} c_s(\omega)^T x_s^i(\omega) \leq t, \\ \Leftrightarrow \quad & \forall \omega \in \Omega, \quad \exists i \in [k] \quad \underbrace{A_s(\omega) x_s^i \leq b(\omega) \text{ and } c_s(\omega)^T x_s^i(\omega) \leq t}_{\Leftrightarrow x_s^i \in P_t(\omega)}. \end{aligned}$$

Hence, the value of Problem (4) is at most t if and only if $P_t(\Omega)$ admits a hitting set of size k . Observe that in particular, Problem (4) is infeasible if and only if there is no feasible solution $(x_s^1, x_s^2, \dots, x_s^k)$, that is no hitting set of size k for $P(\Omega)$. ◀

When there is a first decision ($\ell > 0$), one may proceed as in Lemma 5, *mutatis mutandis*, and obtain the following characterizations of the fact that the value of (3) is at most t :

(a) The affine family of polyhedra $P_t(\Omega) = \{P_t(\omega) : \omega \in \Omega\}$ defined by

$$P_t(\omega) = \left\{ \begin{pmatrix} x_f \\ x_s \end{pmatrix} \in \mathbb{R}^{\ell+\delta} : \begin{pmatrix} A_f(\omega) & A_s(\omega) \\ c_f^T & c_s(\omega)^T \end{pmatrix} \begin{pmatrix} x_f \\ x_s \end{pmatrix} \leq \begin{pmatrix} b(\omega) \\ t \end{pmatrix} \right\}$$

admits a hitting set of size k in which all points have the same first ℓ coordinates.

(b) There exists $x_f \in \mathbb{R}^\ell$ such that the affine family of polyhedra $P_{x_f, t}(\Omega) = \{P_{x_f, t}(\omega) : \omega \in \Omega\}$ defined by

$$P_{x_f, t}(\omega) = \left\{ x_s \in \mathbb{R}^\delta : \begin{pmatrix} A_s(\omega) \\ c_s(\omega)^T \end{pmatrix} x_s \leq \begin{pmatrix} b(\omega) - A_f(\omega) x_f \\ t - c_f^T x_f \end{pmatrix} \right\}$$

admits a hitting set of size k .

In other words, we have to either require the hitting set to lie on some coordinate subspace (a), or work with an affinely parameterized family of affine families of polyhedra (b). Each affine family of polyhedra in (b) corresponds to the slice of the affine family of polyhedra of (a) by a coordinate subspace of codimension ℓ .

2.3 A simpler correspondence via lifting

A reformulation of the k -adaptability problem (3) neater than those of Section 2.2 can be obtained by linearizing the problem via a “lifting” similar to the classical Veronese map. Specifically, let us fix ℓ and p and define

$$L : \begin{cases} \mathbb{R}^{\ell+p} & \rightarrow \mathbb{R}^{\ell+p+\ell p} \\ (x_f, \underbrace{\omega_1, \omega_2, \dots, \omega_p}_{\omega}) & \mapsto (x_f, \omega, \omega_1 x_f, \omega_2 x_f, \dots, \omega_p x_f). \end{cases}$$

Observe that $L(\mathbb{R}^{\ell+p})$ is the surface $\Sigma \subseteq \mathbb{R}^d$ of dimension $\ell + p$ defined by the quadratic equations $z_{p+i\ell+j} = z_j z_{\ell+i}$ for $1 \leq i \leq p$ and $1 \leq j \leq \ell$. Letting $\pi : (z_1, z_2, \dots, z_d) \mapsto (z_1, z_2, \dots, z_{\ell+p})$ be the projection that forgets the last ℓp coordinates, we see that L is a bijection with inverse $\pi|_{\Sigma}$.

We use L to recast any set defined by *bilinear* conditions in $\mathbb{R}^{\ell+p}$ between x_f and ω as the intersection of Σ with a set defined by *linear* conditions in $\mathbb{R}^{\ell+p+\ell p}$. For better readability, we let $d = \ell + p + \ell p$, use z to denote a point in \mathbb{R}^d , and write $z_{x_f} = (z_1, \dots, z_{\ell})^T$ and $z_{\omega} = (z_{\ell+1}, \dots, z_{\ell+p})^T$. Now, for $\omega \in \Omega$, let us decompose $A_f(\omega)$ into $A_f(\omega) = A_{L,0} + \sum_{i \in [p]} \omega_i A_{L,i}$, where $A_{L,0}, A_{L,1}, \dots, A_{L,p} \in \mathbb{R}^{m \times \delta}$. We then define, for every $x_s \in \mathbb{R}^{\delta}$,

$$\check{Q}_t(x_s) = \left\{ z \in \mathbb{R}^d : \begin{cases} A_{L,0} z_{x_f} + A_s(z_{\omega}) x_s + \sum_{i \in [p]} A_{L,i} (z_{ip+\ell+1}, \dots, z_{ip+\ell+p})^T \leq b(z_{\omega}) \\ c_f^T z_{x_f} + c_s(z_{\omega}) x_s \leq t \end{cases} \right\}.$$

Notice that this definition replaces every bilinear term $\omega_i A_{L,i} x_f$ arising in $A_f(\omega) x_f$ by its linear “lift” $A_{L,i} (z_{ip+\ell+1}, \dots, z_{ip+\ell+p})^T$. We also define $\check{P}_t(x_f) = L(x_f \times \Omega)$ for every $x_f \in \mathbb{R}^{\ell}$.

Now, for any fixed $t \in \mathbb{R}$, $\check{P}_t(\mathbb{R}^{\ell}) = \{\check{P}_t(x_f) : x_f \in \mathbb{R}^{\ell}\}$ and $\check{Q}_t(\mathbb{R}^{\delta}) = \{\check{Q}_t(x_s) : x_s \in \mathbb{R}^{\delta}\}$ are affine families of polyhedra. For $\check{Q}_t(\mathbb{R}^{\delta})$, this follows from the assumption that $A_s(\omega)$ depends affinely on ω and $A_{L,0}, A_{L,1}, \dots, A_{L,p}$ are constant matrices. For $\check{P}_t(\mathbb{R}^{\ell})$, this follows from the assumption that Ω is a polyhedron.

► **Lemma 6.** *For any real t , the value of the k -adaptability problem (3) is at most t if and only if there is a member of $\check{P}_t(\mathbb{R}^{\ell})$ that can be covered by some k members of $\check{Q}_t(\mathbb{R}^{\delta})$.*

Proof. Let us fix $t \in \mathbb{R}$ and start from the characterization (b) from Section 2.2 – from which we recover the notation $P_{x_f,t}(\omega)$. The value of the k -adaptability problem (3) is at most t if and only if there exists $(x_f, x_s^1, x_s^2, \dots, x_s^k) \in \mathbb{R}^{\ell} \times (\mathbb{R}^{\delta})^k$ such that for every $\omega \in \Omega$, there exists $i \in [k]$ such that $x_s^i \in P_{x_f,t}(\omega)$. Letting $S_t(x_s) = \{(x_f, \omega) \in \mathbb{R}^{\ell} \times \Omega : x_s \in P_{x_f,t}(\omega)\}$, the characterization becomes: there exists $(x_f, x_s^1, x_s^2, \dots, x_s^k) \in \mathbb{R}^{\ell} \times (\mathbb{R}^{\delta})^k$ such that $\{x_f\} \times \Omega$ is covered by $S_t(x_s^1) \cup S_t(x_s^2) \cup \dots \cup S_t(x_s^k)$.

Let us now consider how the lift L acts on this characterization. First, L is a bijection from $\mathbb{R}^{\ell+p}$ to Σ , so $\{x_f\} \times \Omega$ is covered by $S_t(x_s^1) \cup S_t(x_s^2) \cup \dots \cup S_t(x_s^k)$ if and only if $L(\{x_f\} \times \Omega)$ is covered by $L(S_t(x_s^1)) \cup L(S_t(x_s^2)) \cup \dots \cup L(S_t(x_s^k))$. Now, $L(\{x_f\} \times \Omega) = \check{P}(x_f)$ by definition of \check{P} . On the other hand, the definition of \check{Q} ensures that $L(S_t(x_s)) = \check{Q}_t(x_s) \cap \Sigma$. The characterization therefore reformulates as: there exists $(x_f, x_s^1, x_s^2, \dots, x_s^k) \in \mathbb{R}^{\ell} \times (\mathbb{R}^{\delta})^k$ such that $\check{P}(x_f)$ is covered by $(\check{Q}_t(x_s^1) \cap \Sigma) \cup (\check{Q}_t(x_s^2) \cap \Sigma) \cup \dots \cup (\check{Q}_t(x_s^k) \cap \Sigma)$. Since for every $x_f \in \mathbb{R}^{\ell}$, the set $\check{P}(x_f)$ is contained in Σ , we can drop the intersections with Σ in that characterization, and the statement follows. ◀

3 Polynomial complexity bounds

We first show that algorithms for quantifier elimination yield, when the parameter k and the dimensions are constant, strongly polynomial algorithms for the decision version of the k -adaptability Problem (3) as well as for our continuous hitting and covering problems.

3.1 Semi-algebraic sets and first-order formulas

Our proofs involve the analysis of families of polynomial inequalities in \mathbb{R}^d , that is of semi-algebraic sets. It will be convenient to formulate these sets in the equivalent form of formulas from the first-order theory of the reals with addition and multiplication (see the book of Basu, Pollack and Roy [4, §2.3]). We will use the following complexity bounds on quantifier elimination.

► **Theorem 7** ([4, §14, Theorem 14.16]). *Let $\phi(X_1, X_2, \dots, X_d)$ be a first-order formula of the form $(Q_1 \mathbf{Y}_1)(Q_2 \mathbf{Y}_2) \dots (Q_n \mathbf{Y}_n)F(X, Y)$, where Q_1, Q_2, \dots, Q_n are quantifiers in $\{\forall, \exists\}$ and F is a quantifier free formula. Assume that each set \mathbf{Y}_i has at most b variables and that F is built with at most s polynomials of degree at most r (in $nb + d$ variables).*

- (i) *There exists a quantifier free formula equivalent to $\phi(X_1, X_2, \dots, X_d)$ involving at most ν polynomials, each of degree at most δ ,*
- (ii) *this quantifier free formula can be computed in ν arithmetic operations in the ring generated by the coefficients of the polynomials, and*
- (iii) *if the coefficients of the input polynomials are in \mathbb{Z} and have bitsize at most τ , then the polynomials that arise in the algorithm have coefficients of bitsize bounded by $\tau\delta$,*
where $\nu = s^{(d+1)(b+1)^n} r^{dO(b)^n}$ and $\delta = r^{O(b)^n}$.

We will also use the following complexity bound for the existential theory of the reals.

► **Theorem 8** ([4, §13, Theorem 13.13]). *Let ϕ be a quantifier free formula with d free variables, built on ν polynomials of degree at most δ .*

- (i) *The truth of the sentence $(\exists X \in \mathbb{R}^d, \phi(X))$ can be decided in $\nu^{d+1}\delta^{O(d)}$ arithmetic operations in the ring generated by the coefficients of the polynomials, and*
- (ii) *if the coefficients of the input polynomials are in \mathbb{Z} and their bitsize is bounded by τ , the polynomials that arise in the algorithm have coefficients of bitsize bounded by $\tau\delta^{O(d)}$.*

3.2 Application to k -adaptability

The decision version of Problem (3) asks, given some real number t , whether the optimal value of (3) is at most t .

Proof of Theorem 1. The decision version of Problem 3 can be cast as deciding, given t , the validity of the formula

$$\exists x_f \in \mathbb{R}^\ell, x_s^1, x_s^2, \dots, x_s^k \in \mathbb{R}^\delta \quad \Phi_t(x_f, x_s),$$

where

$$\begin{aligned} \Phi_t(x_f, x_s) \equiv \forall \omega \in \mathbb{R}^p \quad & (\omega \notin \Omega) \vee ((x_f, x_s^1) \in P_t(\omega)) \vee ((x_f, x_s^2) \in P_t(\omega)) \vee \dots \\ & \dots \vee ((x_f, x_s^k) \in P_t(\omega)), \end{aligned}$$

and $P_t(\Omega)$ is the affine family of polyhedra defined by

$$P_t(\omega) = \left\{ (x_f, x_s) \in \mathbb{R}^{\ell+\delta} : \begin{pmatrix} A_f(\omega) & A_s(\omega) \\ c_f(\omega)^T & c_s(\omega)^T \end{pmatrix} (x_f, x_s) \leq \begin{pmatrix} b(\omega) \\ t \end{pmatrix} \right\}.$$

Note that the polyhedron $P_t(\omega)$ encodes both the constraints enforced by A_f, A_s , and b on (x_f, x_s^i) , and the bound t on the optimal value (like in the proof of Lemma 4). The universal quantifier on ω takes care of the $\sup_{\omega \in \Omega}$ in the formulation (3), while the disjunction on the terms of the form $(x_f, x_s^i) \in P_t(\omega)$ takes care of the $\inf_{i \in [k]}$.

We can apply Theorem 7 to eliminate the universal quantifier in $\Phi_t(x_f, x_s)$. Referring to the parameters in Theorem 7, we have $n = 1$ and $b = p$, and there are $s \leq v + k(m + 1)$ polynomials involved, defining the polyhedra Ω and $P_t(\omega)$. Each polynomial is of degree at most $r = 2$, and the total number of free variables is $k\delta + \ell$. Hence, the elimination of the quantifier takes time $O\left((v + m)^{(k\delta + \ell + 1)(p + 1)}\right)$.

We now have a formula with a single existential quantifier on (x_f, x_s) , to which we can apply Theorem 8. Following Theorem 7, the degree is at most $r^{O(b)^n} = 2^{O(p)}$, and we can therefore decide the validity of the formula in time

$$\left(O\left((v + m)^{(k\delta + \ell + 1)(p + 1)}\right)\right)^{k\delta + \ell + 1} 2^{O((k\delta + \ell)p)} = O\left((v + m)^{(k\delta + \ell + 1)^2(p + 1)}\right)$$

as claimed. \blacktriangleleft

Similarly, we can express the hitting and covering problems for affine families of polyhedra as the decision of the validity of a formula, which can then be handled by Theorems 7 and 8. This gives the following results:

- For every constant k, p , and d there is an algorithm that decides in time strongly polynomial in m , given a polyhedron Ω in \mathbb{R}^p defined by m constraints and an affine family of polyhedra $P(\Omega)$ in \mathbb{R}^d , each defined by at most m constraints, whether $P(\Omega)$ admits a hitting set of size k . The complexity is $m^{O(1)}$, where the exponent depends on k, p , and d .
- For every constant k, γ, λ and d there is an algorithm that decides in time strongly polynomial in m , given an affine family of polyhedra $P(\Gamma)$ in \mathbb{R}^d defined by at most m constraints, and an affine family of polyhedra $Q(\Lambda)$ in \mathbb{R}^d , defined by at most m constraints, with $\Gamma \subseteq \mathbb{R}^\gamma$ and $\Lambda \subseteq \mathbb{R}^\lambda$ polyhedra defined by at most m constraints each, whether there exist $\gamma_0 \in \Gamma$ and $\lambda_1, \lambda_2, \dots, \lambda_k \in \Lambda$ such that $P(\gamma_0) \subseteq Q(\lambda_1) \cup Q(\lambda_2) \cup \dots \cup Q(\lambda_k)$. The complexity is $m^{O(1)}$, where the exponent depends on k, γ, λ and d .

This approach can also be applied to the decision version of the two-stage robust optimization problem (2), yielding an algorithm with higher complexity than for the decision version of finite adaptability (3), see [12, Appendix C]. We also note that similar tools [4, §14, Algorithm 14.9] can solve the optimization version of the k -adaptability Problem (3), with Ω a polyhedron in \mathbb{R}^p , in polynomial time for every constant k, ℓ, δ and p .

4 Hitting a one-parameter family of polyhedra

We now prove Theorem 2: for every constant d , the size k of the smallest hitting set of a one-parameter affine family of polytopes in \mathbb{R}^d can be computed in $O(km^2)$, where m denotes the number of constraints defining each polytope. The proof is essentially a greedy algorithm building on an application of parametric search [29] to a carefully chosen linear-time algorithm for linear programming in fixed dimension, here we use Megiddo's [30].

4.1 The intersection of an affine family of polyhedra

We define the *common intersection region* of an affine family of polyhedra $P(\Omega)$ in \mathbb{R}^d as the set $\mathcal{H}_P(\Omega) = \bigcap_{\omega \in \Omega} P(\omega)$. We use some properties of this region. Even though Section 4 focuses on the one-parameter case ($\Omega \subseteq \mathbb{R}$), we find it convenient to establish these property more generally here.

We let $\text{ext}(C)$ denote the set of extreme³ points of a set $C \subset \mathbb{R}^p$, and write \overline{X} for the topological closure of a subset $X \subseteq \mathbb{R}^p$.

► **Lemma 9.** *For any affine family of polyhedra $P(\Omega)$ and for any subset $\Lambda \subseteq \Omega$ of its parameter set, we have (i) $\mathcal{H}_P(\Lambda) = \mathcal{H}_P(\text{conv } \Lambda)$, (ii) $\mathcal{H}_P(\Lambda) = \mathcal{H}_P(\overline{\Lambda})$ and, (iii) if Λ is bounded, $\mathcal{H}_P(\Lambda) = \mathcal{H}_P(\text{ext } \overline{(\text{conv } \Lambda)})$.*

Proof. Observe first that for every subsets $A, B \subseteq \Omega$, if $A \subseteq B$, then $\bigcap_{\omega \in B} P(\omega) \subseteq \bigcap_{\omega \in A} P(\omega)$. It follows that $\mathcal{H}_P(\Lambda) \supseteq \mathcal{H}_P(\text{conv } \Lambda)$ and $\mathcal{H}_P(\Lambda) \supseteq \mathcal{H}_P(\overline{\Lambda})$.

Let us prove that $\mathcal{H}_P(\Lambda) \subseteq \mathcal{H}_P(\text{conv } \Lambda)$. Let $x \in \bigcap_{\omega \in \Lambda} P(\omega)$. Consider a finite convex combination $\omega = \sum_{i=1}^v \alpha_i \lambda_i$ where $v \in \mathbb{N}$ and $\lambda_1, \dots, \lambda_v \in \Lambda$. By assumption, for all $i \in [v]$, x hits the polyhedron $P(\lambda_i)$, that is for all $i \in [v]$, $A(\lambda_i)x \leq b(\lambda_i)$. It follows that

$$A(\omega)x = \sum_{i=1}^v \alpha_i A(\lambda_i)x \leq \sum_{i=1}^v \alpha_i b(\lambda_i) = b(\omega),$$

and $x \in P(\omega)$. This completes the proof of (i).

Let us now prove that $\mathcal{H}_P(\Lambda) \subseteq \mathcal{H}_P(\overline{\Lambda})$. Let $x \in \bigcap_{\omega \in \Lambda} P(\omega)$. Let $\omega \in \overline{\Lambda}$. Then there exists a sequence $(\lambda_n)_{n \in \mathbb{N}}$ of elements of Λ that converges to ω . By assumption, for all $n \in \mathbb{N}$, x hits the polyhedron $P(\lambda_n)$, that is for all $n \in \mathbb{N}$, $A(\lambda_n)x \leq b(\lambda_n)$. Since A and b are continuous, it comes that $A(\omega)x \leq b(\omega)$, that is $x \in P(\omega)$. This completes the proof of (ii).

Let us prove $\mathcal{H}_P(\Lambda) = \mathcal{H}_P(\text{ext } \overline{(\text{conv } \Lambda)})$. A theorem of Minkowski (see [26, Chapter A, § 2.3, Theorem 2.3.4]) asserts that if C is compact, convex in a finite dimensional euclidean space, then C is the convex hull of its extreme points. Hence, if Λ is bounded, $\text{conv } \overline{\Lambda}$ is a compact convex set in \mathbb{R}^p , and is therefore the convex hull of its extreme points. So we have:

$$\begin{aligned} \bigcap_{\omega \in \Lambda} P(\omega) &\stackrel{(i)}{=} \bigcap_{\omega \in \text{conv } \Lambda} P(\omega) \stackrel{(ii)}{=} \bigcap_{\omega \in \overline{\text{conv } \Lambda}} P(\omega) \\ &= \bigcap_{\omega \in \text{conv}(\text{ext } \overline{(\text{conv } \Lambda)})} P(\omega) \stackrel{(i)}{=} \bigcap_{\omega \in \text{ext}(\overline{\text{conv } \Lambda})} P(\omega). \quad \blacktriangleleft \end{aligned}$$

This has an interesting computational consequence, which is already known as a tractable instance of static robust optimization [6, Chapter 1, Corollary 1.3.5 (i)], and which we include here for completeness.

► **Corollary 10.** *Let $\Omega \subseteq \mathbb{R}^p$ be a polytope with v vertices and let $P(\Omega)$ be an affine family of polyhedra in \mathbb{R}^d , with every member of the family defined by m constraints. Deciding the existence of a one-point hitting set for $P(\Omega)$ reduces to solving a linear program with d variables and $v \cdot m$ constraints.*

Proof. We apply Lemma 9 (i) to the set $\Lambda = \text{ext}(\Omega)$. Since $\text{conv}(\text{ext}(\Omega)) = \Omega$, it comes that $\bigcap_{\omega \in \text{ext}(\Omega)} P(\omega) = \bigcap_{\omega \in \text{conv}(\text{ext}(\Omega))} P(\omega) = \bigcap_{\omega \in \Omega} P(\omega)$. That is to say, $P(\Omega)$ admits a one-point hitting set if and only if the set $P(\text{ext}(\Omega)) = \{P(\omega) : \omega \in \text{ext}(\Omega)\}$ admits a one-point hitting set. The latter condition is equivalent to the existence of $x \in \mathbb{R}^d$ such that for all $\omega \in \text{ext}(\Omega)$, $A(\omega)x \leq b(\omega)$. This is equivalent to finding a feasible solution of a linear program with d variables and $v \cdot m$ constraints. ◀

³ Recall that a point c in a convex C is called *extreme* if it cannot be expressed as the midpoint of two distinct points in C , or equivalently if $C \setminus \{c\}$ is convex [26, Chapter A, § 2.3, Definition 2.3.1].

4.2 Structure of a minimum-size hitting set

Let $\alpha, \beta \in \mathbb{R}$ and let $P([\alpha, \beta])$ be an affine family of polyhedra. By Lemma 9, for any point $x \in \mathbb{R}^d$, the set $\hat{P}(x) = \{\omega \in \mathbb{R} : x \in P(\omega)\}$ is a closed interval. For $\lambda \in [\alpha, \beta]$ let us define

$$\sigma_P(\lambda) = \sup \left\{ \lambda \cup \left\{ \nu \in [\alpha, \beta] : \exists x \in \mathbb{R}^d \text{ s. t. } [\lambda, \nu] \subseteq \hat{P}(x) \right\} \right\}.$$

Note that if $P(\lambda) = \emptyset$, then $\sigma_P(\lambda) = \lambda$. Also note that the supremum $\sigma_P(\lambda)$ may be finite but not a maximum, in the sense that $P([\lambda, \sigma_P(\lambda)])$ may not have a one-point hitting set. Indeed, consider for example $[\alpha, \beta] = [0, 2]$ and $P(\omega) = \{x \in \mathbb{R} : x(\omega - 1) \geq 1\}$; For $\omega \in [0, 1)$, we have $P(\omega) = (-\infty, \frac{1}{\omega-1}]$, so that $\sigma_P(0) = 1$; Note, however, that $P(1) = \emptyset$, which prevents $P([0, 1])$ from having a (one-point) hitting set. When $P([\alpha, \beta])$ is an affine family of *polytopes*, we argue that the supremum $\sigma_P(\lambda)$ is always attained.

► **Lemma 11.** *Let $\alpha, \beta \in \mathbb{R}$ and let $P([\alpha, \beta])$ be an affine family of polytopes. For every $\lambda \in [\alpha, \beta]$, either $P(\lambda)$ is empty or there exists $x_\lambda \in \mathbb{R}^d$ such that $\hat{P}(x_\lambda)$ contains $[\lambda, \sigma_P(\lambda)]$.*

Proof. Let $\lambda \in [\alpha, \beta]$. We can assume that $P(\lambda)$ is nonempty, as otherwise the statement is trivial. We can also assume that $\sigma_P(\lambda) > \lambda$, as otherwise any point $x_\lambda \in P(\lambda)$ satisfies the condition. Now, suppose, by contradiction, that $\bigcap_{t \in [\lambda, \sigma_P(\lambda)]} P(t) = \mathcal{H}_P([\lambda, \sigma_P(\lambda)])$ is empty. Since $\sigma_P(\lambda) > \lambda$, we have $\mathcal{H}_P([\lambda, \sigma_P(\lambda)]) = \mathcal{H}_P([\lambda, \sigma_P(\lambda)])$ by Lemma 9 (ii), so $\bigcap_{t \in [\lambda, \sigma_P(\lambda)]} P(t)$ is already empty. Helly's theorem asserts that if a (possibly infinite) family of compact convex sets in \mathbb{R}^d has empty intersection, then some $d + 1$ of them already have empty intersection. Therefore, there exist $t_1 \leq t_2 \leq \dots \leq t_{d+1}$ in $[\lambda, \sigma_P(\lambda)]$ such that $\bigcap_{i=1}^{d+1} P(t_i)$ is empty. Yet, $t_{d+1} < \sigma_P(\lambda)$ ensures that there exists $x \in \mathbb{R}^d$ such that $[\lambda, t_{d+1}] \subseteq \hat{P}(x)$. In particular, $x \in \bigcap_{i=1}^{d+1} P(t_i)$, a contradiction. ◀

Let us write $\sigma_P^{(i)} = \underbrace{\sigma_P \circ \sigma_P \circ \dots \circ \sigma_P}_{i \text{ times}}$. Lemma 11 yields a simple characterization of the size of hitting sets of one-parameter families of polytopes.

► **Lemma 12.** *An affine family of polytopes $P([\alpha, \beta])$ has a hitting set of size k if and only if $\sigma_P^{(k)}(\alpha) = \beta$.*

Proof. Suppose that $\sigma_P^{(k)}(\alpha) = \beta$. Let $t_0 = \alpha$ and for $i = 1, 2, \dots, k$ let $t_i = \sigma_P(t_{i-1})$. By Lemma 11, for $i = 1, 2, \dots, k$ there exists $x_i \in \mathbb{R}^d$ such that $\hat{P}(x_i) \supseteq [t_{i-1}, t_i]$. Since $t_k = \sigma_P^{(k)}(\alpha) = \beta$, we have $\bigcup_{i=1}^k [t_{i-1}, t_i] = [\alpha, \beta]$, and $\{x_1, x_2, \dots, x_k\}$ is a hitting set for $P([\alpha, \beta])$.

Conversely, suppose that $P([\alpha, \beta])$ admits a hitting set $H = \{x_1, x_2, \dots, x_k\}$. Let us write $\hat{P}(x_i) = [\ell_i, r_i]$ and assume, without loss of generality, that $\ell_1 \leq \ell_2 \leq \dots \leq \ell_k$. Observe that, by definition, $r_i = \min\{\beta, \sigma_P(\ell_i)\}$, hence $r_i \leq \sigma_P(\ell_i)$ for all $i \in [k]$. Also observe that $\ell_i \leq r_{i-1}$, since otherwise H misses $P(\omega)$ for some $\omega \in (r_{i-1}, \ell_i)$. We must also have $\beta \leq r_k$, since otherwise H misses $P(\beta)$, and $\ell_1 \leq \alpha$, otherwise H misses $P(\alpha)$. Combining these observations, and using the fact that the function σ_P is nondecreasing, we obtain:

$$\beta \leq r_k \leq \sigma_P(\ell_k) \leq \sigma_P(r_{k-1}) \leq \sigma_P(\sigma_P(\ell_{k-1})) \leq \dots \leq \sigma_P^{(k)}(\ell_1) \leq \sigma_P^{(k)}(\alpha).$$

Since $\sigma_P(\lambda)$ is at most β , we must have $\sigma_P^{(k)}(\alpha) = \beta$, as claimed. ◀

4.3 Parametric search on a linear programming algorithm

Lemma 12 reduces the computation of a minimum-size hitting set for a one-parameter affine family of polytopes $P([\alpha, \beta])$ to the computation of the function σ_P . We now focus on the latter problem.

4.3.1 Comparing is easy

Let us fix some real s and write $s^* = \sigma_P(s)$; we assume that s is known, and want to determine s^* . We first notice that we already have efficient algorithms for comparing any input real t to s^* : this is equivalent to deciding whether $P([s, t])$ can be hit by a single point, and therefore reduces to testing whether $P(s) \cap P(t)$ is empty, by Corollary 10, which can be done by solving a linear program with d variables and $2m$ constraints. Let us note that several deterministic algorithms have been proposed to solve such linear program in $O(m)$ time when the dimension d is fixed [30, 17, 18, 3, 16, 9, 14].

4.3.2 From comparing to computing

It turns out that a general technique, called *parametric search* [29], can turn⁴ an algorithm for *comparing to s^** into an algorithm for *computing s^** , the latter having complexity $O(f(n)^2)$ where $f(n)$ is the complexity of the former. This suggests that all we need is to apply parametric search to one of these linear-time algorithms for linear programming to get a quadratic-time algorithm to compute σ_P . There is, however, a catch: parametric search can only be applied to a comparison algorithm that can be modeled by an algebraic decision tree where the decisions have bounded degree. This is, fortunately, the case of Megiddo's algorithm [30] (a fact more easily checked from Clarkson's summary of that algorithm [17]). For completeness, we summarize in the next subsections the model of algebraic decision trees and the parametric search technique.

4.3.3 Background: algebraic decision trees

The *algebraic decision tree* model of computation [35] represents an algorithm with input $x \in \mathbb{R}^n$ as a rooted tree where (i) each internal node v is decorated with a polynomial p_v with constant coefficients and with variables some of the input x_i , (ii) for each internal node, the edges towards its descendants are labeled by sign conditions (< 0 , ≤ 0 , $= 0$, $\neq 0$, ≥ 0 or > 0) that are mutually exclusive and cover all cases, and (iii) each leaf has a label that serves as answer to the algorithm. To execute an algebraic decision tree T on an input $x \in \mathbb{R}^n$, we traverse T starting from the root, determining at each node v the sign of the polynomial p_v on the input x and taking the corresponding edge towards a descendant, and return the label of the leaf we eventually reach. Let us stress that this model of computation is nonuniform, in the sense that the tree is actually a family of trees, one per input size n . Let us also emphasize that if a polynomial Real RAM algorithm can be modeled by an algebraic decision tree in which every node polynomial has constant degree, then that algorithm is strongly polynomial.

4.3.4 Background: Parametric search

Let $D(\lambda)$ be a decision problem depending on a real parameter λ . Suppose that D is *monotone* in the sense that if $D(\lambda)$ is true, then $D(\mu)$ is also true for all $\mu \leq \lambda$. Then, there exists some $\lambda^* = \sup\{\lambda \in \mathbb{R} : D(\lambda) \text{ is true}\}$, possibly ∞ . Parametric search [29] (see also [2]) is a technique to turn certain algorithms \mathcal{A} for evaluating $\lambda \mapsto P(\lambda)$ into algorithms for computing λ^* . This technique applies to algorithms presented as algebraic decision trees. Intuitively, it consists in identifying the path in \mathcal{A} that would be followed if that tree was executed on λ^* .

⁴ A more effective technique for this purpose was proposed by Chan [13], but unfortunately we could not apply it here.

Let us sketch a simplification of that technique sufficient for our purpose. We traverse the tree while maintaining an interval I known to contain λ^* , initially set to \mathbb{R} . When reaching a node v , we check whether the sign of the polynomial p_v is constant over I . If it is, we follow the suitable branch and keep I unchanged. Otherwise, we compute the set S of roots of p_v and use \mathcal{A} to determine $P(\lambda)$ for each $\lambda \in S$; the monotony ensures that the largest root r of p_v such that $P(r)$ is true and the smallest root s of p_v such that $P(s)$ is false are consecutive in S ; they determine our new interval I , and we follow the branch indicated by the sign of p_v on I . We continue this procedure until we reach a leaf, and return the upper endpoint of I . Altogether, if h is the height of the tree, the parametric search takes $O(h)$ calls to \mathcal{A} on specific values as well as $O(1)$ additional computation at each level. In other words, if \mathcal{A} has complexity $O(f(n))$, then the parametric search determines the exact value of λ^* in time $O(f(n)^2)$.

4.4 Proof of Theorem 2

Let us first summarize how our algorithm works in the real RAM model.

► **Proposition 13.** *For every constant d , there is a Real-RAM algorithm that computes in time $O(km^2)$, given an affine family of polytopes $P([\alpha, \beta])$, each defined by at most m constraints, the size k of the smallest hitting set of $P([\alpha, \beta])$.*

Proof. We are given as input $P([\alpha, \beta])$. We put $\alpha_0 = \alpha$ and compute $\alpha_i = \sigma_P(\alpha_{i-1})$, for $i = 1, 2, \dots$, until we reach some value $\alpha_i \geq \beta$. At this point, we return i as the size of the smallest hitting set of $P([\alpha, \beta])$. We compute α_i from α_{i-1} by performing parametric search on the algorithm \mathcal{A} that, given some real t , uses Megiddo's algorithm to solve the linear program that determines whether $t \leq \sigma_P(\alpha_{i-1})$. Since the dimension d is fixed, Megiddo's algorithm takes $O(m)$ time to solve one comparison to $\sigma_P(\alpha_{i-1})$, and the computation of one α_i takes $O(m^2)$ time. Altogether, the algorithm takes $O(km^2)$ time, where k is the size of the smallest hitting set of $P([\alpha, \beta])$. ◀

To prove Theorem 2, it remains to analyze the bit complexity of the numbers manipulated by the algorithm of Proposition 13 when k is fixed in addition to the ambient dimension d of the polytopes. This proof relies on the observation that if \mathcal{A} is an algebraic decision tree of height h that decides a monotone decision problem $P(\cdot)$, then the algorithm obtained by performing parametric search on \mathcal{A} can also be modeled by a tree of height $O(h^2)$, which is almost an algebraic decision tree (except for the leaves, that do not output only true or false). The key ingredient for this is the following lemma.

► **Lemma 14.** *For every constant d_1 and d_2 , there exist algebraic decision trees of constant complexity that, given an integer $r \leq d_1$ and a vector $V \in \mathbb{R}^{d_1+d_2+2}$ describing the coefficients of two univariate polynomials $p_1(X)$ and $p_2(X)$ of degree at most d_1 and d_2 , respectively, solve the following problems:*

- (i) *does $p_1(X)$ have at least r real roots?*
- (ii) *is the value of p_2 in the r -th real root of p_1 strictly positive (resp. strictly negative, zero)?*

Proof. We first reformulate questions (i) and (ii) as formulas from the first-order theory of the reals in a constant number of variables. Let $\pi_s(V, X)$ denote $p_s(X)$ for $s = 1, 2$. For (i), the polynomial p_1 has at least r distinct real roots ($\alpha_1 < \dots < \alpha_r$) if and only if the following

formula is true:

$$\exists(x_1, x_2, \dots, x_{d_1}) \in \mathbb{R}^{d_1}, \left(\bigwedge_{\ell=1}^{d_1} (\pi_1(V, x_\ell) = 0 \vee (\ell > r)) \right) \wedge \left(\bigwedge_{\ell=1}^{d_1-1} (x_{\ell+1} > x_\ell) \right).$$

For (ii), we treat the case of strictly positive sign, the two other cases follow the same path. Observe that the sign of $\pi_2(V, \cdot)$ in the r th root of $\pi_1(V, \cdot)$ is positive if and only if

- the polynomial $x \mapsto \pi_1(V, x)$ has at least r distinct real roots $x_1 < x_2 < \dots < x_r$,
- any other root of $x \mapsto \pi_1(V, x)$ is larger than x_r , and
- $\pi_2(V, x_r) > 0$.

This holds if and only if the following formula is true (we have r appear in the formula only as a value, not as an index, to ensure that eventually the tree depends only on d_1 and d_2):

$$\begin{aligned} & \exists(x_1, x_2, \dots, x_{d_1}) \in \mathbb{R}^{d_1}, \forall x \in \mathbb{R}, \\ & \left(\bigwedge_{\ell=1}^{d_1} (\pi_1(V, x_\ell) = 0 \vee (\ell > r)) \right) \wedge \left(\bigwedge_{\ell=1}^{d_1-1} (x_{\ell+1} > x_\ell) \right) \\ & \wedge \left((\pi_1(V, x) = 0) \Rightarrow \left(\bigvee_{\ell=1}^{d_1} ((x = x_\ell) \wedge (\ell \leq r)) \right) \vee \left(\bigwedge_{\ell=1}^{d_1} ((x > x_\ell) \vee (\ell > r)) \right) \right) \\ & \wedge \bigwedge_{\ell=1}^{d_1} (\pi_2(V, x_\ell) > 0 \vee (\ell \neq r)). \end{aligned}$$

By Theorem 7, such a formula is equivalent to a quantifier free formula in the input vector V and r , involving a constant number of polynomials of constant degree. That quantifier-free formula can in turn be modeled by an algebraic decision tree of constant complexity. ◀

We can now complete the proof of Theorem 2.

Proof of Theorem 2. Let V be a vector of real numbers consisting of the parameters that represent $P([\alpha, \beta])$. Let \mathcal{A}_0 denote an algebraic decision tree that models the comparison, using Megiddo's linear-time algorithm, of t to $\sigma_P(s)$. The input of \mathcal{A}_0 is s , V and t . The height of \mathcal{A}_0 is $O(m)$.

Now, consider the real RAM algorithm that computes $\sigma_P(s)$ by performing parametric search on \mathcal{A}_0 . This algorithm has complexity $O(m^2)$ and takes s , V and t as input. Observe that the value that it outputs is not an arbitrary real: the principles of parametric search ensure that the output is either $+\infty$ or a real root of a polynomial Π_1 determining the branching in one of the nodes of \mathcal{A}_0 (see Section 4.3.4). We can therefore represent the output of the parametric search by an integer r_1 (to mean we are interested in the r_1 th real root) and a polynomial of constant degree and whose coefficients are constant degree polynomials of some input coordinates. We can model this parametric search by a tree \mathcal{A}_1 that is almost an algebraic decision tree: \mathcal{A}_1 takes s and V as input, and each leaf contains the description of $\sigma_P(s)$ in the form of the integer r_1 and the polynomial Π_1 . The only aspect in which \mathcal{A}_1 is not an algebraic decision tree is that it does not solve a decision problem.

We can now append at each leaf of \mathcal{A}_1 a second copy of \mathcal{A}_1 where the input parameter s has been substituted for the r_1 th root of Π_1 . This only requires changing, in the second copy of \mathcal{A}_1 , for every branching polynomial Π , the one-node evaluation of the sign of Π in s by the constant-size subtree evaluating the sign of Π in the r_1 th root of Π_1 given by Lemma 14. This results in a tree \mathcal{A}_2 that takes s and V as input, and where each leaf contains the description of $\sigma_P^{(2)}(s)$. Again, the principles of parametric search ensures that the output

can be presented in the form of an integer r_2 and a polynomial Π_2 of constant degree, the coefficients of which are constant degree polynomials of some input coordinates. Again, the only aspect in which \mathcal{A}_2 is not an algebraic decision tree is that it does not solve a decision problem. Again, \mathcal{A}_2 has height $O(m^2)$.

And so on, for every constant i we can construct a tree \mathcal{A}_i of height $O(m^2)$, that takes s and V as input, and where each leaf contains the description of $\sigma_P^{(i)}(s)$. Finally, in \mathcal{A}_k , we substitute every leaf by a constant-size algebraic decision tree that compares the number stored in that leaf to the input number β . The resulting tree \mathcal{A} is an algebraic decision tree of height $O(m^2)$ that takes s and V as input, and that compares $\sigma_P^{(k)}(s)$ to β . Executing this tree for $s = \alpha$ gives a strongly quadratic algorithm for deciding whether $P([\alpha, \beta])$ has a k -point hitting set. ◀

Proving Corollary 3 now amounts to going from a decision problem to the associated optimization problem.

Proof of Corollary 3. With Lemma 5 and Theorem 2, we already have that for every constant d and k , there is an algorithm that solves the decision version of the k -adaptability Problem (3) in the special case where $\ell = 0$ and Ω is an interval, in time strongly quadratic in m . That algorithm can be modeled by an algebraic decision tree (this is the final tree \mathcal{A} in the proof of Theorem 2). We can therefore perform parametric search on that algorithm to solve the optimization version of these problems in time strongly quartic in m . ◀

5 Concluding remarks

1. The quantifier elimination algorithms do not need the maps A_f , A_s , b , and c_s (for the finite adaptability) or the maps A and b (for families of polyhedra) to be affine, but merely to be polynomial of constant degree. Theorem 1 thus generalizes, *mutatis mutandis*, to this setting. Note that the assumption that these maps are affine is used in the correspondence between the finite adaptability and the hitting/covering of affine families of polyhedra, in the hitting/covering duality, and in the proof of Theorem 2 and Corollary 3.
2. The assumption in Theorem 2 that P be an affine family of polytopes, rather than polyhedra, is used to ensure that σ_P is not only a supremum, but actually a maximum (this underlies the proof of Lemma 11). We conjecture that if P is a one-parameter family of polyhedra and $P([\lambda, \sigma_P(\lambda)])$ cannot be hit by a single point, then the polyhedron $P(\sigma_P(\lambda))$ must be empty. If that is true, then Theorem 2 readily generalizes to affine families of polyhedra.
3. A natural question is whether the hitting set problem for affine families of polyhedra is fixed-parameter tractable with respect to the dimensions (of the ambient space of the polyhedra as well as the parameter space).

References

- 1 Pankaj K. Agarwal. Range searching. In *Handbook of discrete and computational geometry*, pages 1057–1092. Chapman and Hall/CRC, 2017.
- 2 Pankaj K. Agarwal and Micha Sharir. Efficient algorithms for geometric optimization. *ACM Comput. Surv.*, 30(4):412–458, 1998. doi:10.1145/299917.299918.
- 3 Pankaj K. Agarwal, Micha Sharir, and Sivan Toledo. An efficient multi-dimensional searching technique and its applications. Technical report, Duke University, USA, 1993.

- 4 Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in real algebraic geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer, 2006. 2nd ed. URL: <https://hal.science/hal-01083587>.
- 5 A Ben-Tal, L El Ghaoui, and A Nemirovski. Robust optimization—princeton university press, 2009.
- 6 Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*, volume 28 of *Princeton Series in Applied Mathematics*. Princeton University Press, 2009. doi:10.1515/9781400831050.
- 7 Aharon Ben-Tal and Arkadi Nemirovski. Robust optimization—methodology and applications. *Mathematical programming*, 92:453–480, 2002. doi:10.1007/S101070100286.
- 8 Dimitris Bertsimas and Constantine Caramanis. Finite adaptability in multistage linear optimization. *IEEE Trans. Autom. Control.*, 55(12):2751–2766, 2010. doi:10.1109/TAC.2010.2049764.
- 9 Hervé Brönnimann, Bernard Chazelle, and Jirí Matoušek. Product range spaces, sensitive sampling, and derandomization. *SIAM J. Comput.*, 28(5):1552–1575, 1999. doi:10.1137/S0097539796260321.
- 10 Hervé Brönnimann and Michael T. Goodrich. Almost optimal set covers in finite VC-dimension. *Discret. Comput. Geom.*, 14(4):463–479, 1995. doi:10.1007/BF02570718.
- 11 Christoph Buchheim and Jannis Kurtz. Min-max-min robust combinatorial optimization. *Math. Program.*, 163(1-2):1–23, 2017. doi:10.1007/S10107-016-1053-Z.
- 12 Jean Cardinal, Xavier Goaoc, and Sarah Wajsbrot. Hitting and covering affine families of convex polyhedra, with applications to robust optimization, 2025. doi:10.48550/arXiv.2504.16642.
- 13 Timothy M. Chan. Geometric applications of a randomized optimization technique. *Discret. Comput. Geom.*, 22(4):547–567, 1999. doi:10.1007/PL00009478.
- 14 Timothy M. Chan. Improved deterministic algorithms for linear programming in low dimensions. *ACM Trans. Algorithms*, 14(3):30:1–30:10, 2018. doi:10.1145/3155312.
- 15 Bernard Chazelle, Herbert Edelsbrunner, Michelangelo Grigni, Leonidas J. Guibas, Micha Sharir, and Emo Welzl. Improved bounds on weak epsilon-nets for convex sets. *Discret. Comput. Geom.*, 13:1–15, 1995. doi:10.1007/BF02574025.
- 16 Bernard Chazelle and Jirí Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. *J. Algorithms*, 21(3):579–597, 1996. doi:10.1006/JAGM.1996.0060.
- 17 Kenneth L. Clarkson. Linear programming in $O(n3^{d^2})$ time. *Inf. Process. Lett.*, 22(1):21–24, 1986. doi:10.1016/0020-0190(86)90037-2.
- 18 Martin E. Dyer. On a multidimensional search technique and its application to the euclidean one-centre problem. *SIAM J. Comput.*, 15(3):725–738, 1986. doi:10.1137/0215052.
- 19 Khaled Elbassioni. A bicriteria approximation algorithm for the minimum hitting set problem in measurable range spaces. *Oper. Res. Lett.*, 51(5):507–514, 2023. doi:10.1016/J.ORL.2023.07.005.
- 20 Jeff Erickson. Concrete models of computation, 2003. URL: <https://jeffe.cs.illinois.edu/teaching/497/06-algebraic-tree.pdf>.
- 21 Jeff Erickson, Ivor van der Hoog, and Tillmann Miltzow. Smoothing the gap between NP and ER. *SIAM J. Comput.*, 53(6):S20–102, 2024. doi:10.1137/20M1385287.
- 22 Guy Even, Dror Rawitz, and Shimon Shahar. Hitting sets when the VC-dimension is small. *Inf. Process. Lett.*, 95(2):358–362, 2005. doi:10.1016/J.IPL.2005.03.010.
- 23 Grani A Hanasusanto, Daniel Kuhn, and Wolfram Wiesemann. K-adaptability in two-stage robust binary programming. *Operations Research*, 63(4):877–891, 2015. doi:10.1287/OPRE.2015.1392.
- 24 Grani Adiwena Hanasusanto, Daniel Kuhn, and Wolfram Wiesemann. K-adaptability in two-stage robust binary programming. *Oper. Res.*, 63(4):877–891, 2015. doi:10.1287/OPRE.2015.1392.

- 25 David Haussler and Emo Welzl. epsilon-nets and simplex range queries. *Discret. Comput. Geom.*, 2:127–151, 1987. doi:10.1007/BF02187876.
- 26 Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of Convex Analysis*. Springer Verlag, Heidelberg, 2001.
- 27 Safia Kedad-Sidhoum, Anton Medvedev, and Frédéric Meunier. Finite adaptability in two-stage robust optimization: asymptotic optimality and tractability. arXiv 2305.05399, 2023. arXiv:2305.05399.
- 28 Maarten Löffler. *Data Imprecision in Computational Geometry*. PhD thesis, Utrecht University, Netherlands, 2009. URL: <http://dspace.library.uu.nl/handle/1874/36022>.
- 29 Nimrod Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *J. ACM*, 30(4):852–865, 1983. doi:10.1145/2157.322410.
- 30 Nimrod Megiddo. Linear programming in linear time when the dimension is fixed. *J. ACM*, 31(1):114–127, 1984. doi:10.1145/2422.322418.
- 31 Nimrod Megiddo. On the complexity of some geometric problems in unbounded dimension. *J. Symb. Comput.*, 10(3/4):327–334, 1990. doi:10.1016/S0747-7171(08)80067-3.
- 32 Nimrod Megiddo and Kenneth J. Supowit. On the complexity of some common geometric location problems. *SIAM J. Comput.*, 13(1):182–196, 1984. doi:10.1137/0213014.
- 33 Krzysztof Postek and Dick den Hertog. Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. *INFORMS J. Comput.*, 28(3):553–574, 2016. doi:10.1287/IJOC.2016.0696.
- 34 Steve Smale. Mathematical problems for the next century. *The Mathematical Intelligencer*, 20(2):7–15, 1998. doi:10.1007/BF03025291.
- 35 J. Michael Steele and Andrew Chi-Chih Yao. Lower bounds for algebraic decision trees. *J. Algorithms*, 3(1):1–8, 1982. doi:10.1016/0196-6774(82)90002-5.
- 36 Anirudh Subramanyam, Chrysanthos E. Gounaris, and Wolfram Wiesemann. K -adaptability in two-stage mixed-integer robust optimization. *Math. Program. Comput.*, 12(2):193–224, 2020. *Extended version:* <https://arxiv.org/abs/1706.07097>. doi:10.1007/S12532-019-00174-2.
- 37 Ivor van der Hoog, Irina Kostitsyna, Maarten Löffler, and Bettina Speckmann. Preprocessing imprecise points for the pareto front. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 3144–3167. SIAM, 2022. doi:10.1137/1.9781611977073.122.
- 38 Ivor van der Hoog, Irina Kostitsyna, Maarten Löffler, and Bettina Speckmann. Sorting under partial (interval order) information. *J. Comput. Geom.*, 15(1):143–171, 2024. doi:10.20382/JOCG.V15I1A6.
- 39 Ihsan Yanikoglu, Bram L. Gorissen, and Dick den Hertog. A survey of adjustable robust optimization. *Eur. J. Oper. Res.*, 277(3):799–813, 2019. doi:10.1016/J.EJOR.2018.08.031.