



Approximations of unbounded convex projections and unbounded convex sets

Gabriela Kováčová¹ · Birgit Rudloff²

Received: 17 October 2023 / Accepted: 4 December 2024 / Published online: 13 January 2025
© The Author(s) 2025

Abstract

We consider the problem of projecting a convex set onto a subspace or, equivalently formulated, the problem of computing a set obtained by applying a linear mapping to a convex feasible set. This includes the problem of approximating convex sets by polyhedrons. The existing literature on convex projections provides methods for bounded convex sets only, in this paper we propose a method that can handle both bounded and unbounded problems. The algorithms we propose build on the ideas of inner and outer approximation. In particular, we adapt the recently proposed methods for solving unbounded convex vector optimization problems to handle also the class of projection problems.

Keywords Unbounded convex sets · Convex projection · Inner and outer approximation

1 Introduction

In this work, we consider a convex projection problem with the focus on the unbounded case. Our interest in the unbounded convex projection problem arises from the appearance of potentially unbounded convex projections in the implementation of the set-valued Bellman principle. The set-valued Bellman principle is a counterpart of the famous Bellman principle for non-standard problems, such as problems with multiple objectives, see e.g. [9], ratios of two objectives [11] or set-valued objectives, see e.g. [6, 7]. Convex projections have also been applied to solve reachable sets of control systems, see [16].

The aim is to compute (or approximate) a set obtained by projecting elements of a convex feasible set. The problem formulation appearing in [10, 16] is to

$$\text{compute } \{y \in \mathbb{R}^k \mid \exists z \in \mathbb{R}^m : (z, y) \in S\}, \quad (1.1)$$

for a given convex feasible set $S \subseteq \mathbb{R}^{m+k}$. Note that the problem is equivalent to computing (or approximating) a convex set, see Lemma 3.1.

✉ Birgit Rudloff
birgit.rudloff@wu.ac.at
Gabriela Kováčová
kovacova@ucla.edu

¹ Department of Mathematics, University of California, Los Angeles, Los Angeles, CA 90095, USA

² Institute for Statistics and Mathematics, Vienna University of Economics and Business, 1020 Vienna, Austria

Problem (1.1) with a polyhedral feasible set S is known as a polyhedral projection and was studied in [13]. The authors of [13] proved an equivalence between polyhedral projection, multi-objective linear programming and vector linear programming. As a consequence, a polyhedral projection (1.1) can be solved via an associated multi-objective linear program

$$\text{minimize } \begin{pmatrix} y \\ -\mathbb{1}^\top y \end{pmatrix} \text{ w.r.t. } \leq_{\mathbb{R}_+^{k+1}} \text{ subject to } (z, y) \in S. \quad (1.2)$$

This inspired subsequent research also for the convex case. [16] considers a convex problem (1.1) with a compact feasible set and proposes an algorithm to solve the problem. Their algorithm adapts the primal algorithm of [12] applied to the multi-objective problem (1.2). [10] directly examines the relation between convex problems (1.1) and (1.2). The equivalence between boundedness properties of the two problems is shown and their corresponding solutions are related in the bounded (and self-bounded) case, see Sect. 3 for precise definitions. A polyhedral approximation of a convex body through the associated multi-objective problem is considered in [14]. The authors derive error bounds in dependence of the stopping criterion of the algorithm for the multi-objective problem.

The aim of this paper is to provide a solution method for the unbounded case, which has not been treated in the existing literature so far. We start, however, by revisiting the bounded case. The existing methods solve a bounded convex projection, implicitly or explicitly, through the associated multi-objective problem. In consequence, computations are done in the image space of the multi-objective problem (1.2) which has a higher dimension than the image space of the projection problem (1.1). We propose an algorithm for solving a bounded projection problem working directly in the image space of the problem. The algorithm is based on an inner and outer approximation of the target set; these ideas have been used for solving convex vector optimization problems, see e.g. [1, 12], as well as for approximation of convex bodies, see e.g. [2, 8]. The proposed algorithm can be seen as an alternative to existing methods for the bounded case that does not require an increase of the dimension. It is also used as a part of the proposed solution method in the unbounded case.

In case of unbounded convex projections, also the associated multi-objective problem is unbounded. A method for solving unbounded multi-objective convex problems has only recently been introduced in [18]. In this paper we adapt the ideas of the algorithms of [18] and propose a method for solving unbounded convex projections. The problem of approximating convex sets by polyhedrons was recently studied in [4, 5], where algorithms were developed for the particular case of spectrahedra.

2 Preliminaries

For a set $X \subseteq \mathbb{R}^n$ we denote the closure of X , the interior of X , the relative interior of X , the convex hull of X and the convex conic hull of X by $\text{cl } X$, $\text{int } X$, $\text{relint } X$, $\text{conv } X$ and $\text{cone } X$, respectively. We use the convention $\text{cone } \emptyset := \{0\}$. The *recession cone* of a set $X \subseteq \mathbb{R}^n$ is

$$X_\infty = \{d \in \mathbb{R}^n : x + \lambda d \in X \quad \forall x \in X, \lambda \geq 0\}.$$

Elements of the recession cone are called *recession directions*. A cone $X \subseteq \mathbb{R}^n$ is called *solid* if $\text{int } X \neq \emptyset$.

Every *polyhedron* has an *H-representation* and a *V-representation*. In the *H-representation* a polyhedron $X \subseteq \mathbb{R}^n$ is given as a finite intersection of half-spaces

$$X = \bigcap_{i=1}^k \{x \in \mathbb{R}^n : w_i^\top x \leq \alpha_i\},$$

where $k \in \mathbb{N}$, $w_1, \dots, w_k \in \mathbb{R}^n \setminus \{0\}$ and $\alpha_1, \dots, \alpha_k \in \mathbb{R}$. In the *V-representation* a polyhedron $X \subseteq \mathbb{R}^n$ is determined by points $v_1, \dots, v_{k_v} \in \mathbb{R}^n$ and directions $d_1, \dots, d_{k_d} \in \mathbb{R}^n \setminus \{0\}$, for $k_v \in \mathbb{N} \setminus \{0\}$ and $k_d \in \mathbb{N}$, as

$$X = \text{conv}\{v_1, \dots, v_{k_v}\} + \text{cone}\{d_1, \dots, d_{k_d}\}.$$

We assume that the set of points as well as the set of directions do not contain any redundant elements. In the sequel, we denote by $\text{vert}X$ the set of points $v_1, \dots, v_{k_v} \in \mathbb{R}^n$ determining the V-representation of X . Note that when the polyhedron does not contain any line, the points $\text{vert}X$ are the vertices of X and the V-representation is uniquely determined. For a polyhedron containing a line we consider an arbitrary (non-redundant) V-representation.

We primarily use the l_1 (Manhattan) norm, denoted by $\|\cdot\| := \|\cdot\|_1$, to measure distances. When the l_2 (Euclidean) norm is used it is denoted by $\|\cdot\|_2$. A closed ϵ -ball around the origin (in the l_1 -norm) is denoted by B_ϵ . The main reason for the choice of the l_1 -norm is the fact (utilized within Algorithm 6.1) that the associated ball is a polyhedron. The distance between two sets $X, Y \subseteq \mathbb{R}^n$ is measured as their *Hausdorff distance*

$$d_H(X, Y) = \max \left\{ \sup_{x \in X} \inf_{y \in Y} \|x - y\|, \sup_{y \in Y} \inf_{x \in X} \|x - y\| \right\},$$

where we use the convention $d_H(\emptyset, \emptyset) := 0$. Note that $d_H(X, Y) \leq \epsilon$ for closed sets X and Y implies $X \subseteq Y + B_\epsilon$ and $Y \subseteq X + B_\epsilon$. The *diameter* of a set $X \subseteq \mathbb{R}^n$ is defined as

$$\text{diam}(X) = \sup\{\|x - y\| : x, y \in X\}.$$

3 Problem formulation and solution concepts

Throughout the paper we will use the problem formulation of computing a set obtained by applying a linear mapping onto a convex feasible set. This is an equivalent formulation of the convex projection problem, as Lemma 3.1 below shows. The formulation was chosen as it requires fewer pieces of notation. This section introduces the problem, its properties and appropriate solution concepts.

We will denote by $\mathbb{X} \subseteq \mathbb{R}^n$ a convex feasible set and by $A \in \mathbb{R}^{a \times n}$ a matrix. The problem of interest is

$$\text{compute } A[\mathbb{X}] = \{Ax : x \in \mathbb{X}\}. \quad (\text{P})$$

Since the set $A[\mathbb{X}]$ does not need to be closed, even if the feasible set $\mathbb{X}$ is closed, we will throughout the paper work with the closed image set

$$\mathcal{A} = \text{cl } A[\mathbb{X}].$$

Lemma 3.1 *Problem (P) and problem (1.1) are equivalent.*

Proof Problem (1.1) can be formulated in the form of problem (P) for the particular choice of $a = k$, $n = m + k$, $\mathbb{X} = S$ and $A = (0_{k \times m}, I_{k \times k})$. Problem (P) can be formulated in the form

of problem (1.1) for the particular choice of $m = n$, $k = a$ and $S = \{(x, y) \in \mathbb{R}^{n+a} \mid x \in \mathbb{X}, y = Ax\}$. $\square$

The following summarizes the assumptions made throughout this paper.

Assumption 3.2 $\mathbb{X} \subseteq \mathbb{R}^n$ is a convex set with a non-empty interior, $\text{int } \mathbb{X} \neq \emptyset$. $A \in \mathbb{R}^{a \times n}$ is a matrix. They jointly determine the problem (P).

The boundedness properties of problem (P) determine which solution concepts and solution methods are appropriate, as one can already see in [10, 16] and is also discussed below. We now provide a definition of a bounded problem and shortly discuss afterwards various notions of the more general concept of self-boundedness that one can encounter in the literature. Then, we recall the solution concept for bounded problems and propose a solution concept for the unbounded case.

Definition 3.3 Problem (P) is **bounded** if $\mathcal{A}$ is a bounded set. Otherwise, it is **unbounded**.

Consider the following notions of *self-boundedness* for a closed convex set $\mathcal{A} \subseteq \mathbb{R}^a$:

- (S1) $\mathcal{A} \subseteq \{a^0\} + \mathcal{A}_\infty$ for some $a^0 \in \mathbb{R}^a$.
- (S2) $\mathcal{A} \subseteq \text{conv}\{a^1, \dots, a^k\} + \mathcal{A}_\infty$ for some $k \in \mathbb{N}$, $a^1, \dots, a^k \in \mathbb{R}^a$.
- (S3) $\sup_{x \in \mathcal{A}} \inf_{y \in \mathcal{A}_\infty} \|x - y\| < \infty$.

The notion (S1) was introduced in [17] for (an upper image of) a vector optimization problem. (S2) was proposed in [10] as a generalization of (S1) appropriate for a convex projection problem. (S3) appears in [5] as a notion appropriate for general (line-free) convex sets, the concept originates from [15]. These three properties are closely related, as the next lemma summarizes.

Lemma 3.4 Consider a closed convex set $\mathcal{A} \subseteq \mathbb{R}^a$. It holds $(S1) \Rightarrow (S2) \Leftrightarrow (S3)$. If the recession cone $\mathcal{A}_\infty$ is solid, then all three notions are equivalent.

Proof $(S1) \Rightarrow (S2)$ is obvious. Let us show $(S2) \Leftrightarrow (S3)$:

- Assume (S2) and take any $x \in \mathcal{A}$. Then $x = \sum \alpha^i a^i + y$ for some $y \in \mathcal{A}_\infty$ and $\alpha^i \geq 0$ with $\sum \alpha^i = 1$. We get $\|x - y\| = \|\sum \alpha^i a^i\| \leq \max_{i=1, \dots, k} \|a^i\|$, which proves (S3).
- Assume (S3) and set $K := \sup_{x \in \mathcal{A}} \inf_{y \in \mathcal{A}_\infty} \|x - y\| < \infty$. This means that for every $x \in \mathcal{A}$ there exists $y \in \mathcal{A}_\infty$ such that $x - y \in B_K$, therefore $\mathcal{A} \subseteq \mathcal{A}_\infty + B_K$. Since the polyhedral ℓ_1 ball B_K is a convex hull of finitely many points, this proves (S2).

[5] showed $(S1) \Leftrightarrow (S3)$ for a set with a solid $\mathcal{A}_\infty$. $\square$

Finally, we discuss solution concepts for the considered problem. What we understand under *solving* (P) is approximating the set $\mathcal{A}$ (and hence also $A[\mathbb{X}]$) in some (appropriate) sense. The following definition of a solution that is appropriate for a bounded problem is taken from [10].

Definition 3.5 A non-empty finite set $\bar{X} \subseteq \mathbb{X}$ is a **finite ϵ -solution** of a bounded problem (P) if it holds

$$\text{conv } A[\bar{X}] + B_\epsilon \supseteq \mathcal{A}.$$

[10] shows that such a finite ϵ -solution exists for an arbitrary tolerance $\epsilon > 0$ for a bounded convex projection problem. An approach for finding such solutions through solving the associated multi-objective problems is also provided—we shortly recapped this in the introduction. However, this solution concept is clearly not appropriate once we drop the assumption of boundedness—an unbounded set $\mathcal{A}$ cannot be reasonably approximated by a bounded polyhedron $\text{conv } A[\bar{X}] + B_\epsilon$. This is true even for unbounded problems that are self-bounded.

For this reason, we propose a more general notion of approximate solution here that is appropriate for bounded as well as unbounded problems (regardless of being self-bounded or not). It is based on the notion of a finite (ϵ, δ) -solution of a convex vector optimization problem considered in [18]. The main idea is for the solution to contain, alongside a set of feasible points, also a set of directions to approximate the recession cone of the image set.

Definition 3.6 A pair $(\bar{X}, \mathcal{Y})$ is a **finite (ϵ, δ) -solution** of problem (P) if $\emptyset \neq \bar{X} \subseteq \mathbb{X}$ and $\mathcal{Y} \subseteq \mathbb{R}^a$ are finite sets that satisfy

1. $d_H(\mathcal{A}_\infty \cap B_1, \text{cone } \mathcal{Y} \cap B_1) \leq \delta$,
2. $\mathcal{A}_\infty \subseteq \text{cone } \mathcal{Y}$, and
3. $\text{conv } A[\bar{X}] + \text{cone } \mathcal{Y} + B_\epsilon \supseteq \mathcal{A}$.

The above notion of a finite (ϵ, δ) -solution (as well as the counterpart in [18] for convex vector optimization problems) is closely related to the notion of a (polyhedral) outer approximation of a set introduced in [5]. We state this definition here as well with one modification—throughout this work we use the Manhattan norm instead of the Euclidean norm.

Definition 3.7 [5, Definition 4.2] Given a nonempty, closed convex and line-free set $C \subseteq \mathbb{R}^a$, a line-free polyhedron P is called an (ϵ, δ) -outer approximation of C if

1. $d_H(P_\infty \cap B_1, C_\infty \cap B_1) \leq \delta$,
2. $P \supseteq C$,
3. $\sup_{p \in \text{vert } P} \inf_{c \in C} \|p - c\| \leq \epsilon$.

The following relation holds between Definitions 3.6 and 3.7: A finite (ϵ, δ) -solution $(\bar{X}, \mathcal{Y})$ of (P) generates a polyhedron $\text{conv } A[\bar{X}] + \text{cone } \mathcal{Y} + B_\epsilon$ that is an (ϵ, δ) -outer approximation of the set $\mathcal{A}$. For a bounded problem, a finite ϵ -solution $\bar{X}$ yields a finite $(\epsilon, 0)$ -solution $(\bar{X}, \emptyset)$ in the sense of Definition 3.6 and it also generates a polyhedron $\text{conv } A[\bar{X}] + B_\epsilon$ that is an $(\epsilon, 0)$ -outer approximation of the bounded set $\mathcal{A}$.

4 Scalarizations of (P)

The previous works provide a way to solve a (bounded) problem (P) through the associated multi-objective problem. These approaches, such as [16, Algorithm 3.3], work with scalarizations of the associated multi-objective problem. In this work we will construct algorithms to solve the problem (P) directly, without (implicitly or explicitly) transforming it into a multi-objective problem. Therefore, we will need scalarizations of problem (P). This section introduces three scalarizations—the weighted sum, the Pascoletti-Serafini and the norm minimization—and provides some related results. Parallel results for scalarizations of a convex vector optimization problem can be found in [1, 12].

The **weighted sum scalarization** of (P), given a weight $w \in \mathbb{R}^a \setminus \{0\}$, is

$$\min_{x \in \mathbb{X}} w^\top Ax. \quad (P_1(w))$$

Lemma 4.1 *If $(P_1(w))$ is bounded and x^w is the optimal solution, then $\{y \in \mathbb{R}^a : w^\top y \geq w^\top Ax^w\} \supseteq \mathcal{A}$.*

The **Pascoletti-Serafini scalarization** of (P) with point $v \in \mathbb{R}^a$ and direction $d \in \mathbb{R}^a \setminus \{0\}$ is

$$\max_{(x, \alpha) \in \mathbb{X} \times \mathbb{R}} \alpha \quad \text{s.t.} \quad Ax = v + \alpha d. \quad (P_2(v, d))$$

Its dual problem is

$$\min_{\lambda \in \mathbb{R}^a} \left\{ -\lambda^\top v + \sup_{x \in \mathbb{X}} \{\lambda^\top Ax\} \right\} \quad \text{s.t.} \quad \lambda^\top d = 1. \quad (D_2(v, d))$$

Remark 4.2 Strong duality between $(P_2(v, d))$ and its dual $(D_2(v, d))$ does not hold in general. As a counterexample one can consider $\mathbb{X} = \{x \in \mathbb{R}^2 | x_1 > 0, x_2 \geq 0\} \cup \{x \in \mathbb{R}^2 | x_1 = 0, 0 \leq x_2 \leq 1\}$, the matrix A being the identity, the point $v = (0, 0)$ and the direction $d = (0, 1)$. Then, the primal problem $(P_2(v, d))$ is bounded with optimal value 1, but the dual problem $(D_2(v, d))$ is unbounded. In general, the existence of a pair (x, α) strictly feasible for the primal problem is sufficient for strong duality, see [3, Section 5.2.3]. Therefore, strong duality holds when the point v is chosen from the relative interior of the set $A[\mathbb{X}]$.

Lemma 4.3 *Let $v \in \text{relint } A[\mathbb{X}]$ and let problem $(P_2(v, d))$ be bounded. Assume that (x^*, α^*) and λ^* are optimal solutions of $(P_2(v, d))$ and $(D_2(v, d))$, respectively. Then it holds $\{y \in \mathbb{R}^a : (\lambda^*)^\top y \leq (\lambda^*)^\top Ax^*\} \supseteq \mathcal{A} \supseteq A[\mathbb{X}]$.*

Proof Strong duality and feasibility give

$$\sup_{x \in \mathbb{X}} \{(\lambda^*)^\top Ax\} = (\lambda^*)^\top v + \alpha^* = (\lambda^*)^\top (Ax^* - \alpha^* d) + \alpha^* = (\lambda^*)^\top Ax^*.$$

This implies $(\lambda^*)^\top Ax \leq (\lambda^*)^\top Ax^*$ for all $x \in \mathbb{X}$ and the result follows. $\square$

Lemma 4.4 *If the problem $(P_2(v, d))$ is unbounded, then d is a recession direction of $\mathcal{A}$.*

Proof Since $\mathcal{A}$ is a closed convex set, to prove that $d \in \mathcal{A}_\infty$ it suffices to show that for some $v^0 \in \mathcal{A}$ it holds

$$v^0 + \alpha d \in \mathcal{A} \quad \forall \alpha \geq 0.$$

Let us first find a point $v^0 \in \mathcal{A}$: Unboundedness of the problem $(P_2(v, d))$ implies that there is a pair of $x^0 \in \mathbb{X}$ and $\alpha^0 \geq 0$ feasible for the problem, so $v^0 := Ax^0 = v + \alpha^0 d \in A[\mathbb{X}] \subseteq \mathcal{A}$.

Now let $\alpha \geq 0$. Since $(P_2(v, d))$ is unbounded, there exists a feasible $\alpha^1 \geq \alpha^0 + \alpha$ with a corresponding $x^1 \in \mathbb{X}$, which gives us $v + \alpha^1 d = Ax^1 \in A[\mathbb{X}]$. Since $v^0 + (\alpha^1 - \alpha^0)d = v + \alpha^1 d$, convexity of the set $A[\mathbb{X}]$ implies $v^0 + \alpha d \in A[\mathbb{X}] \subseteq \mathcal{A}$. $\square$

Lemma 4.5 *If $v \in A[\mathbb{X}]$, then $(P_2(v, d))$ is feasible.*

The **norm minimization scalarization** of (P) given a point $v \in \mathbb{R}^a$ is

$$\min_{(x,z) \in \mathbb{X} \times \mathbb{R}^a} \|z\|_2 \quad \text{s.t.} \quad Ax = v + z. \quad (P_3(v))$$

Its dual problem is

$$\max_{\lambda \in \mathbb{R}^a} \left\{ -\lambda^\top v + \inf_{x \in \mathbb{X}} \lambda^\top Ax \right\} \quad \text{s.t.} \quad \|\lambda\|_2 \leq 1. \quad (D_3(v))$$

To derive this consider the dual function

$$q(\lambda) = \inf_{x \in \mathbb{X}, z \in \mathbb{R}^a} L(x, z, \lambda) = -\lambda^\top v + \inf_{x \in \mathbb{X}} \lambda^\top Ax + \inf_{z \in \mathbb{R}^a} \{\|z\|_2 - \lambda^\top z\},$$

where the last term corresponds to the conjugate of $\|z\|_2$, see [3, Example 3.26].

The Norm minimization scalarization was used by [1] in an algorithm for solving convex vector optimization problems. Unlike in the rest of the paper, we used the ℓ_2 norm here. Strong duality, boundedness and existence of a supporting hyperplane derived below hold for the scalarization with all choices of a norm. Using the ℓ_2 norm helps to avoid possible problems in the case of non-uniqueness of an optimal solution under the ℓ_1 norm, even though in numerical experiments both norms worked equally well. The scalarization will be utilized in Algorithm 5.1.

Lemma 4.6 *Problem $(P_3(v))$ is feasible and bounded for any point $v \in \mathbb{R}^a$. Furthermore, there is no duality gap between problems $(P_3(v))$ and $(D_3(v))$.*

Proof Consider an arbitrary $\bar{x} \in \text{int } \mathbb{X}$ and the corresponding $\bar{z} = A\bar{x} - v$. The pair $(\bar{x}, \bar{z})$ is strictly feasible for the problem. The objective function is bounded from below by 0. This yields feasibility, boundedness and strong duality. $\square$

Lemma 4.7 *Assume that (x^*, z^*) and λ^* are optimal solutions of $(P_3(v))$ and $(D_3(v))$, respectively. Then it holds $\{y \in \mathbb{R}^a : (\lambda^*)^\top y \geq (\lambda^*)^\top Ax^*\} \supseteq \mathcal{A} \supseteq A[\mathbb{X}]$.*

Proof From strong duality we have

$$\|z^*\|_2 = -(\lambda^*)^\top v + \inf_{x \in \mathbb{X}} (\lambda^*)^\top Ax = -(\lambda^*)^\top (Ax^* - z^*) + \inf_{x \in \mathbb{X}} (\lambda^*)^\top Ax.$$

We rearrange this expression and add the term $0 = \inf_{z \in \mathbb{R}^a} \{\|z\|_2 - (\lambda^*)^\top z\}$, which originates from the Fenchel conjugate of the norm and dual feasibility, see [3, Example 3.26]. We obtain

$$(\lambda^*)^\top Ax^* - \inf_{x \in \mathbb{X}} (\lambda^*)^\top Ax = \inf_{z \in \mathbb{R}^a} \{\|z\|_2 - (\lambda^*)^\top z\} - (\|z^*\|_2 - (\lambda^*)^\top z^*) \leq 0.$$

Therefore, for all $x \in \mathbb{X}$ it holds $(\lambda^*)^\top Ax^* \leq (\lambda^*)^\top Ax$. $\square$

Remark 4.8 Above we derived half-spaces containing the set $\mathcal{A}$ based on optimal solutions of the three scalarizations and their duals. However, an optimal solution need not be available—it might not even exist. In practice, these scalarizations will be solved by some available solver up to the computational precision of the solver. Let us restate the results of Lemmas 4.1, 4.3 and 4.7 for near-optimal solutions. Here ε denotes the precision.

Firstly, consider a near-optimal solution x^ε of the weighted sum scalarization $(P_1(w))$. One can show that it holds

$$\mathcal{A} \subseteq \{y \in \mathbb{R}^a : w^\top y \geq w^\top Ax^\varepsilon - \varepsilon\}.$$

Secondly, consider near-optimal solutions $(x^\varepsilon, \alpha^\varepsilon)$ and λ^ε of the Pascoletti-Serafini scalarization $(P_2(v, d))$ and its dual $(D_2(v, d))$. One can show that it holds

$$\mathcal{A} \subseteq \{y \in \mathbb{R}^a : (\lambda^\varepsilon)^\top y \leq (\lambda^\varepsilon)^\top Ax^\varepsilon + \varepsilon\}.$$

Thirdly, consider near-optimal solutions $(x^\varepsilon, z^\varepsilon)$ and λ^ε of the norm-minimization scalarization $(P_3(v))$ and its dual $(D_3(v))$. One can show that it holds

$$\mathcal{A} \subseteq \{y \in \mathbb{R}^a : (\lambda^\varepsilon)^\top y \geq (\lambda^\varepsilon)^\top Ax^\varepsilon - \varepsilon\}.$$

Hence, the halfspaces are slightly shifted if only near-optimal solutions of the scalarizations are computed. They are, however, not tilted. This will in particular be important as the recession directions searched within the algorithm proposed in Sect. 6 are not affected by the use of optimal or near-optimal solutions. For more details on the effect of near optimal solutions on the proposed algorithms see Remarks 5.5 and 6.8.

5 Algorithm for a bounded problem (P)

First, we address the case of a bounded problem (P). Such problems can be solved via the associated multi-objective problem by the already available algorithm [16, Algorithm 3.3] or results of [10]. However, doing so runs an algorithm in a $(a + 1)$ -dimensional space. Here we propose an algorithm that works directly in the space $\mathbb{R}^a$. Furthermore, it will be used as part of the algorithm for unbounded problems proposed in Sect. 6.

Our algorithm is based on the idea of finding an outer approximation of the set and iteratively improving the approximation till a desired tolerance level is achieved. As such, Algorithm 5.1 is similar to outer approximation schemes for convex bodies, see [2] for a survey, and to the (primal) algorithms of [1, 12] for bounded convex vector optimization problems (CVOPs). Unlike a CVOP, problem (P) does not involve an ordering cone. Therefore, to obtain an initial outer approximation of the set $\mathcal{A}$, we use the weighted sum scalarizations for standard basis vectors as well as their negatives. This will provide a box-type initial approximation. Afterwards we improve the approximation by applying the norm minimizing scalarization on the vertices of the outer approximation.

Algorithm 5.1 Algorithm for solving a bounded problem (P) satisfying Assumption 3.2.

Input

Bounded problem (P) with a feasible set $\mathbb{X}$ and a matrix A ; tolerance $\epsilon > 0$.

Initialization

1. Set $\bar{X} := \emptyset$, $\mathcal{A}_0 := \mathbb{R}^a$, $V^\epsilon := \emptyset$ and $H := \emptyset$
2. For $w = e^1, \dots, e^a, -e^1, \dots, -e^a$ solve $P_1(w)$ and obtain an optimal solution x^w .

- Set $\bar{X} := \bar{X} \cup \{x^w\}$ and $\mathcal{A}_0 := \mathcal{A}_0 \cap \{y \in \mathbb{R}^a : w^\top y \geq w^\top Ax^w\}$.

Iteration

3. While $H \neq \mathbb{R}^a$
 - (a) Set $H := \mathbb{R}^a$
 - (b) For each $v \in (\text{vert } \mathcal{A}_0) \setminus V^\epsilon$

- Solve problems $P_3(v)$ and $D_3(v)$ to obtain optimal solutions (x^v, z^v) and λ^v .
 - Update $\bar{X} := \bar{X} \cup \{x^v\}$
 - If $\|z^v\| > \epsilon$, then update $H := H \cap \{y \in \mathbb{R}^a : (\lambda^v)^\top y \geq (\lambda^v)^\top A x^v\}$.
Else $V^\epsilon := V^\epsilon \cup \{v\}$.
- (c) Update $\mathcal{A}_0 := \mathcal{A}_0 \cap H$

Output

Set of feasible points $\bar{X}$ and outer approximation $\mathcal{A}_0$ of $\mathcal{A}$.

Remark 5.2 Alternatively, we could initialize the algorithm with solving the weighted sum scalarization ($P_1(w)$) for $w = e^1, \dots, e^a, -\mathbb{1}$ in Step 2. This would also provide a bounded set $\mathcal{A}_0$. We compare the two initializations in Example 7.1, where the alternative version requires fewer operations (optimizations and evaluations of a polyhedron). This is, however, not true for all problems.

In the following we prove some results about the behaviour of Algorithm 5.1. These will be afterwards used to prove the correctness of the proposed algorithm.

Lemma 5.3 *Let the problem (P) be (feasible and) bounded. The following holds for Algorithm 5.1:*

1. Each scalar problem ($P_1(w)$) considered in Step 2 is bounded.
2. After Initialization (Steps 1 and 2), the set $\mathcal{A}_0$ is a bounded convex polyhedron that contains the set $\mathcal{A}$.
3. At each iteration, the set $\mathcal{A}_0$ is a (bounded) convex polyhedral superset of $\mathcal{A}$.

Proof 1. For a bounded problem (P) it holds $A[\mathbb{X}] \subseteq B_K$, respectively $\mathcal{A} \subseteq B_K$, for some K . This implies that scalarization ($P_1(w)$) is bounded for all weights $w \in \mathbb{R}^a \setminus \{0\}$.

2. The set $\mathcal{A}_0$ is initialized as $\mathbb{R}^a$ and throughout Step 2 it is updated via intersections of halfspaces, which contain the set $\mathcal{A}$, see Lemma 4.1. Therefore, $\mathcal{A}_0$ is a convex polyhedral superset of $\mathcal{A}$. Since the set $\mathcal{A}_0$ consists of intersections of halfspaces of the form $\{y \in \mathbb{R}^a : y_i \geq \bar{\alpha}_i\}$ and $\{y \in \mathbb{R}^a : y_i \leq \underline{\alpha}_i\}$ for some $\bar{\alpha}_i, \underline{\alpha}_i \in \mathbb{R}$ and for $i = 1, \dots, a$, it is bounded. As it contains the nonempty set $\mathcal{A}$, it cannot be empty.

3. Within each iteration, set $\mathcal{A}_0$ is updated through intersections with half-spaces that contain the set $\mathcal{A}$, see Lemmas 4.6 and 4.7.

□

Theorem 5.4 *If Algorithm 5.1 terminated successfully for a bounded problem (P), then $\bar{X}$ is a finite ϵ -solution of (P). Furthermore, $\text{conv } A[\bar{X}] + B_\epsilon$ (as well as $\mathcal{A}_0$) is an $(\epsilon, 0)$ -outer approximation of the set $\mathcal{A}$.*

Proof Assume that Algorithm 5.1 terminated successfully for a bounded problem (P). According to Lemma 5.3 and convexity of $\mathcal{A}$ it holds

$$\text{conv } A[\bar{X}] \subseteq \mathcal{A} \subseteq \mathcal{A}_0.$$

The algorithm terminates when

$$\forall v \in \text{vert } \mathcal{A}_0 \quad \exists \bar{x} \in \bar{X} : \quad \|v - A\bar{x}\| \leq \epsilon.$$

Convexity of the involved polyhedrons and convexity of the norm give

$$d_H(\mathcal{A}_0, \text{conv } A[\bar{X}]) \leq \epsilon,$$

which implies

$$\text{conv } A[\tilde{X}] \subseteq \mathcal{A} \subseteq \mathcal{A}_0 \subseteq \text{conv } A[\tilde{X}] + B_\epsilon.$$

□

Remark 5.5 Within Algorithm 5.1 we assume that the optimal solutions of the two scalarizations are found. As stated in Remark 4.8, this will not be the case when the scalarizations are solved numerically (even if optimal solutions exist). Instead, near-optimal solutions will be found, up to a solver accuracy ε . This will result in the cutting half-spaces being shifted (however not tilted) by ε , see Remark 4.8. One could incorporate this quantity into the implementation of the algorithm, but usually $\varepsilon \ll \epsilon$.

Remark 5.6 The correctness of Algorithm 5.1 is proven under the assumption that it terminates, similar to the results for the Benson-type algorithms [12, Algorithm 1] and [1, Algorithm 1] for bounded CVOPs. Finite termination of all these algorithms remains an open question. The authors of [1] proved termination in a finite number of iteration for a modification of [1, Algorithm 2]. If a guarantee of a finite termination is desired for a bounded convex projection, we could apply [1, Algorithm 2] to a counterpart of problem (1.2). However, this would come at the cost of solving the problem in $(a + 1)$ -dimensional space and a difficulty in extending the approach to the unbounded case.

6 Algorithm for a general problem (P)

Now we address problem (P) without the assumption of boundedness. Our aim is again to formulate an algorithm for solving this problem—that is, for finding a finite (ϵ, δ) -solution $(\tilde{X}, \mathcal{Y})$ of (P). Our approach is related in spirit to the method for solving a (possibly unbounded) convex vector optimization problem from [18].

Below we introduce two algorithms: Algorithm 6.1 searches for directions $\mathcal{Y}$ that generate an outer approximation of the recession cone $\mathcal{A}_\infty$ up to a desired tolerance δ . Algorithm 6.5 combines Algorithm 6.1 (as an initialization phase) with the iterations of Algorithm 5.1 to also find the set of feasible points $\tilde{X}$, which together with $\mathcal{Y}$ provide a finite (ϵ, δ) -solution of (P).

Algorithm 6.1 is based on [18, Algorithm 5.2]. However, unlike a CVOP our problem does not contain an ordering cone, therefore no recession direction is known in advance. After initialization, our algorithm iteratively visits directions from the recession cone of the current outer approximation while searching for a recession direction of the set $\mathcal{A}$. If a recession direction is found, it serves as an initial inner approximation of $\mathcal{A}_\infty$, afterwards the inner approximation can be used to iteratively improve the outer one. However, a recession direction need not be found in finitely many steps (think of $\mathcal{A}_\infty$ that is not solid). Then the algorithm terminates once the outer approximation becomes ‘thin’ enough. In both cases, a finite (ϵ, δ) -solution is found.

Algorithm 6.1 Algorithm for approximating the recession cone of $\mathcal{A}$.

Input

Feasible set $\mathbb{X}$ and matrix A ; tolerance $\delta > 0$, point $v \in \text{relint } A[\mathbb{X}]$.

Initialization

1. Set $\tilde{X} := \emptyset$, $\mathcal{A}_0 := \mathbb{R}^a$, $\mathcal{Y}_{in} := \emptyset$, $\Delta := \emptyset$

2. Check boundedness:

For $w = e^1, \dots, e^a, -e^1, \dots, -e^a$ solve problem $P_1(w)$

- If $P_1(w)$ is bounded and x^w is an optimal solution, then update $\bar{X} := \bar{X} \cup \{x^w\}$ and $\mathcal{A}_0 := \mathcal{A}_0 \cap \{y \in \mathbb{R}^a : w^\top y \geq w^\top A x^w\}$.

3. Compute $\mathcal{Y}_{out} := \text{vert}((\mathcal{A}_0)_\infty \cap B_1) \setminus \{0\}$.**Iteration**

4. Search for a recession direction:

While $\mathcal{Y}_{in} = \emptyset$ and $\text{diam}(\mathcal{Y}_{out}) > \delta$

(a) Set $H := \mathbb{R}^a$

(b) For each $d \in \mathcal{Y}_{out} \cup \left\{ \frac{\sum_{d' \in \mathcal{Y}_{out}} d'}{\|\sum_{d' \in \mathcal{Y}_{out}} d'\|} \right\}$ solve problem $P_2(v, d)$

- If $P_2(v, d)$ is unbounded, then update $\mathcal{Y}_{in} := \mathcal{Y}_{in} \cup \{d\}$.
- If (x^d, α^d) and λ^d are optimal solutions of $P_2(v, d)$ and $D_2(v, d)$, then update $\bar{X} := \bar{X} \cup \{x^d\}$ and $H := H \cap \{y \in \mathbb{R}^a : (\lambda^d)^\top y \leq (\lambda^d)^\top A x^d\}$

(c) Update $\mathcal{A}_0 := \mathcal{A}_0 \cap H$ and $\mathcal{Y}_{out} := \text{vert}((\mathcal{A}_0)_\infty \cap B_1) \setminus \{0\}$

5. While $\mathcal{Y}_{out} \setminus (\mathcal{Y}_{in} \cup \Delta) \neq \emptyset$ and $\text{diam}(\mathcal{Y}_{out}) > \delta$

(a) Set $H := \mathbb{R}^a$

(b) For each $d \in \mathcal{Y}_{out} \setminus (\mathcal{Y}_{in} \cup \Delta)$ find $r^d = \arg \min\{\|d - r\| : r \in \mathcal{Y}_{in}\}$

- If $\|d - r^d\| \leq \delta$, then update $\Delta := \Delta \cup \{d\}$

- If $\|d - r^d\| > \delta$, then select $\beta \in (0, 1)$ and solve problem $P_2(v, d^\beta)$ for

$$d^\beta = \frac{\beta d + (1-\beta)r^d}{\|\beta d + (1-\beta)r^d\|}$$

- If $P_2(v, d^\beta)$ is unbounded, then update $\mathcal{Y}_{in} := \mathcal{Y}_{in} \cup \{d^\beta\}$

- If (x^β, α^β) and λ^β are optimal solutions of $P_2(v, d^\beta)$ and $D_2(v, d^\beta)$, then update $\bar{X} := \bar{X} \cup \{x^\beta\}$ and $H := H \cap \{y \in \mathbb{R}^a : (\lambda^\beta)^\top y \leq (\lambda^\beta)^\top A x^\beta\}$

(c) Update $\mathcal{A}_0 := \mathcal{A}_0 \cap H$ and $\mathcal{Y}_{out} := \text{vert}((\mathcal{A}_0)_\infty \cap B_1) \setminus \{0\}$

Output

Set of feasible points $\bar{X}$ and outer approximation $\mathcal{A}_0$ of $\mathcal{A}$.

Inner and outer approximations $\mathcal{Y}_{in}$ and $\mathcal{Y}_{out}$ of $\mathcal{A}_\infty$.

Remark 6.2 In Algorithm 6.1, the point v from the relative interior of $A[\mathbb{X}]$ is treated as an input parameter. Such point can be constructed by performing a feasibility check by means of a phase I stage of an interior-point method, see [3, Section 11.4].

The following lemma provides results that help to prove the correctness of Algorithm 6.1 in Theorem 6.7 below.

Lemma 6.3 Assume that $v \in \text{relint } A[\mathbb{X}]$. The following holds:

1. Each problem $(P_2(v, d))$ solved in Algorithm 6.1 is feasible and strong duality holds for it.
2. In each step of Algorithm 6.1, the set $\mathcal{Y}_{in}$ (if non-empty) contains only normalized recession directions of $\mathcal{A}$.
3. In each step of Algorithm 6.1, the set $\mathcal{A}_0$ is a convex polyhedral superset of $\mathcal{A}$.
4. If the problem (P) is bounded, then Algorithm 6.1 terminates with $\mathcal{Y}_{in} = \mathcal{Y}_{out} = \emptyset$.

Proof 1. This follows from Lemma 4.5 and Remark 4.2.

2. A direction d is added to the set $\mathcal{Y}_{in}$ only when the problem $P_2(v, d)$ is unbounded, all these directions are normalized. The claim follows from Lemma 4.4.
3. The set $\mathcal{A}_0$ is initialized as $\mathbb{R}^d$. It is updated through intersections with supporting halfspaces of $\mathcal{A}$, see Lemmas 4.1 and 4.3.
4. For a bounded problem all weighted sum scalarizations solved in Step 2 are bounded. Therefore, the set $\mathcal{A}_0$ is bounded after Step 2 and $\mathcal{Y}_{out} = \emptyset$ is obtained in Step 3. No iterations of Step 4 or Step 5 are run.

□

Theorem 6.4 *If Algorithm 6.1 terminated successfully for problem (P), then it holds*

$$\text{cone } \mathcal{Y}_{in} \subseteq \mathcal{A}_\infty \subseteq \text{cone } \mathcal{Y}_{out}. \quad (6.1)$$

and

$$d_H(\text{cone } \mathcal{Y}_{out} \cap B_1, \mathcal{A}_\infty \cap B_1) \leq \delta. \quad (6.2)$$

Hence, $\text{cone } \mathcal{Y}_{out}$ is a $(0, \delta)$ -outer approximation of $\mathcal{A}_\infty$ in the sense of Definition 3.7.

Proof If (P) is bounded, Algorithm 6.1 terminates with $\mathcal{Y}_{in} = \mathcal{Y}_{out} = \emptyset$ and the properties are satisfied by definition. Let us consider an unbounded (P). Relation (6.1) follows from convexity, Lemma 6.3 and the fact that $\text{cone } \mathcal{Y}_{out} = (\mathcal{A}_0)_\infty$. Algorithm 6.1 can terminate in two possible ways:

1. The set $\mathcal{Y}_{in}$ is non-empty (i.e. a recession direction was found in Step 4) and it holds $\mathcal{Y}_{out} \subseteq \mathcal{Y}_{in} \cup \Delta$.
2. It holds $\text{diam}(\mathcal{Y}_{out}) \leq \delta$.

We consider these two cases separately and show that each one implies (6.2).

1. In the first case, the termination condition can be stated as

$$\forall d \in \mathcal{Y}_{out} \quad \exists r \in \mathcal{Y}_{in} : \|d - r\| \leq \delta. \quad (6.3)$$

We prove (6.2) by showing that (6.3) implies

$$d_H(\text{cone } \mathcal{Y}_{out} \cap B_1, \text{cone } \mathcal{Y}_{in} \cap B_1) \leq \delta. \quad (6.4)$$

Given the way the sets are constructed, it holds $\text{cone } \mathcal{Y}_{out} \cap B_1 = \text{conv}(\mathcal{Y}_{out} \cup \{0\})$ and $\text{cone } \mathcal{Y}_{in} \cap B_1 = \text{conv}(\mathcal{Y}_{in} \cup \{0\})$. Take any $d \in \text{cone } \mathcal{Y}_{out} \cap B_1$, then $d = \sum_{i=1}^k \alpha^i d^i$ for

some $k \in \mathbb{N}$, $d^1, \dots, d^k \in \mathcal{Y}_{out}$ and $\alpha^1, \dots, \alpha^k \geq 0$ with $\sum_{i=1}^k \alpha^i \leq 1$. According to (6.3),

for each d^i there exists $r^i \in \mathcal{Y}_{in}$ with $\|d^i - r^i\| \leq \delta$. Convexity of the (Manhattan) norm implies $\|\sum_{i=1}^k \alpha^i r^i\| \leq \sum_{i=1}^k \alpha^i \|r^i\| \leq 1$ and $\|d - \sum_{i=1}^k \alpha^i r^i\| \leq \sum_{i=1}^k \alpha^i \|d^i - r^i\| \leq \delta$. This

proves (6.2) since $\sum_{i=1}^k \alpha^i r^i \in \text{cone } \mathcal{Y}_{in} \cap B_1$.

2. Let's consider the second case. Convexity of the norm and $\text{diam}(\mathcal{Y}_{out}) \leq \delta$ imply $\text{diam}(\text{conv } \mathcal{Y}_{out}) \leq \delta$. Recall that $\text{cone } \mathcal{Y}_{out} \cap B_1 = \text{conv}(\mathcal{Y}_{out} \cup \{0\}) = \text{conv}(\text{conv } \mathcal{Y}_{out} \cup \{0\})$. Let us now prove (6.2). Take any $y \in \text{cone } \mathcal{Y}_{out} \cap B_1$, then there exists $\bar{y} \in \text{conv } \mathcal{Y}_{out}$ and $\alpha \in [0, 1]$ such that $y = \alpha \bar{y} + (1 - \alpha)0$. Since $\mathcal{A}_\infty \cap \text{conv } \mathcal{Y}_{out} \neq \emptyset$ and $\text{diam}(\text{conv } \mathcal{Y}_{out}) \leq \delta$, there exists $\bar{a} \in \mathcal{A}_\infty \cap \text{conv } \mathcal{Y}_{out} \subseteq B_1$ such that $\|\bar{y} - \bar{a}\| \leq \delta$. Scaling up yields $\|y - \alpha \bar{a}\| \leq \delta$ for $\alpha \bar{a} \in \mathcal{A}_\infty \cap B_1$, which proves (6.2).

□

Now we state an algorithm for solving a (bounded or unbounded) problem (P). It uses Algorithm 6.1 as an initialization phase and combines it with the iterations of Algorithm 5.1.

Algorithm 6.5 Algorithm for solving problem (P).

Input

A feasible set $\mathbb{X}$ and a matrix A ; tolerances $\epsilon, \delta > 0$, point $v \in \text{relint } A[\mathbb{X}]$.

Initialization

1. Run Algorithm 6.1 with tolerance $\delta > 0$ and point $v \in \text{relint } A[\mathbb{X}]$ to obtain a set of feasible points $\tilde{X}$ and an outer approximation $\mathcal{A}_0$ of $\mathcal{A}$.
2. Set $H := \emptyset$ and $V^\epsilon := \emptyset$

Iteration

3. While $H \neq \mathbb{R}^a$
 - (a) Set $H := \mathbb{R}^a$
 - (b) For each $v \in (\text{vert } \mathcal{A}_0) \setminus V^\epsilon$
 - Solve problems $P_3(v)$ and $D_3(v)$ to obtain optimal solutions (x^v, z^v) and λ^v .
 - Update $\tilde{X} := \tilde{X} \cup \{x^v\}$
 - If $\|z^v\| > \epsilon$, then update $H := H \cap \{y \in \mathbb{R}^a : (\lambda^v)^T y \geq (\lambda^v)^T A x^v\}$.
 - Else $V^\epsilon := V^\epsilon \cup \{v\}$.
 - (c) Update $\mathcal{A}_0 := \mathcal{A}_0 \cap H$

Finalization

4. Compute $\mathcal{Y}_{out} := \text{vert } ((\mathcal{A}_0)_\infty \cap B_1) \setminus \{0\}$

Output

Set of feasible points $\tilde{X}$ and sets of directions $\mathcal{Y}_{in}, \mathcal{Y}_{out}$.
Outer approximation $\mathcal{A}_0$ of $\mathcal{A}$.

The following lemma and theorem prove the correctness of the algorithm.

Lemma 6.6 Assume that $v \in \text{relint } A[\mathbb{X}]$. For Algorithm 6.5 the following holds:

1. The set $\mathcal{A}_0$ is a convex polyhedral superset of $\mathcal{A}$ throughout the algorithm.
2. If problem (P) is bounded, then Algorithm 6.5 returns the same sets $\tilde{X}$ and $\mathcal{A}_0$ as Algorithm 5.1.
3. The recession cone $(\mathcal{A}_0)_\infty$ is a finite $(0, \delta)$ -outer approximation of the recession cone $\mathcal{A}_\infty$ throughout the algorithm.

Proof 1. Algorithm 6.1 returns a convex polyhedral superset of $\mathcal{A}$, see Lemma 6.3. We only update $\mathcal{A}_0$ through intersections with halfspaces containing the set $\mathcal{A}$, see Lemma 4.7.
2. For a bounded problem, only Step 2 of Algorithm 6.1 is relevant (no iterations of Steps 4 and 5 are done). Since Step 2 of Algorithm 6.1 coincides with Step 2 of Algorithm 5.1, the same outer approximation $\mathcal{A}_0$ and the same set of feasible points $\tilde{X}$ are obtained by both. Step 3 of Algorithm 5.1 and Step 3 of Algorithm 6.5 also coincide, yielding the same sets $\tilde{X}$ and $\mathcal{A}_0$.

3. Within Algorithm 6.1 it holds $\text{cone } \mathcal{Y}_{out} = (\mathcal{A}_0)_\infty$. The claim holds at the end of Step 1 of Algorithm 6.5 by Theorem 6.4. Let us denote $\mathcal{A}_0^1, \mathcal{A}_0^2, \dots$ the new sets obtained throughout Algorithm 6.5. Since these sets are obtained via intersections with supporting halfspaces, it holds $\mathcal{A} \subseteq \mathcal{A}_0^i \subseteq \mathcal{A}_0$ for each i . This implies $(\mathcal{A})_\infty \subseteq (\mathcal{A}_0^i)_\infty \subseteq (\mathcal{A}_0)_\infty$, which proves the claim. $\square$

Theorem 6.7 Assume that Algorithm 6.5 terminated successfully for problem (P). Then $(\bar{X}, \mathcal{Y}_{out})$ is a finite (ϵ, δ) -solution of (P).

Proof Let's consider the sets $\bar{X}, \mathcal{Y}_{out}$ and $\mathcal{A}_0$ upon termination of the algorithm. According to Lemma 6.6, $\text{cone } \mathcal{Y}_{out} = (\mathcal{A}_0)_\infty$ is a finite $(0, \delta)$ -outer approximation of the recession cone $\mathcal{A}_\infty$. Therefore, properties 1 and 2 of Definition 3.6 are satisfied. Lemma 6.6 also provides that $\mathcal{A}_0$ is a convex polyhedron. As such, it has a (non-redundant) V-representation $\mathcal{A}_0 = \text{conv}\{v^1, \dots, v^k\} + (\mathcal{A}_0)_\infty$, where $k \in \mathbb{N}$. If $(\mathcal{A}_0)_\infty$ contains no lines, then $v^1, \dots, v^k$ are vertices of $\mathcal{A}_0$. Otherwise these are (non-redundant) points on the boundary of $\mathcal{A}_0$. The iterations of Step 3 of Algorithm 6.5 terminate when for each point v^i there exists $\bar{x}^i \in \bar{X}$ such that $\|v^i - A\bar{x}^i\| \leq \epsilon$. Therefore, it holds $\text{conv}\{v^1, \dots, v^k\} \subseteq \text{conv } A[\bar{X}] + B_\epsilon$. This provides the last property of the solution concept as

$$\mathcal{A} \subseteq \mathcal{A}_0 \subseteq \text{conv } A[\bar{X}] + \text{cone } \mathcal{Y}_{out} + B_\epsilon.$$

$\square$

Algorithm 6.1 provides us with a set of points $\bar{X}$ as well as two sets of directions, $\mathcal{Y}_{in}$ and $\mathcal{Y}_{out}$. It also outputs a polyhedron $\mathcal{A}_0$. Theorems 6.4 and 6.7 provide the inclusions

$$\text{conv } A[\bar{X}] + \text{cone } \mathcal{Y}_{in} \subseteq \mathcal{A} \subseteq \mathcal{A}_0 \subseteq \text{conv } A[\bar{X}] + \text{cone } \mathcal{Y}_{out} + B_\epsilon. \quad (6.5)$$

Let us first discuss the left-hand side inclusion. In the following, we will refer to $\text{conv } A[\bar{X}] + \text{cone } \mathcal{Y}_{in}$ as an inner approximation. Inclusion (6.5) provides an intuition for this choice of terminology. Let us support this intuition a bit more. In case of a non-empty set $\mathcal{Y}_{in}$, equations (6.1) and (6.4) imply

1. $d_H(\mathcal{A}_\infty \cap B_1, \text{cone } \mathcal{Y}_{in} \cap B_1) \leq \delta$,
2. $\mathcal{A}_\infty \supseteq \text{cone } \mathcal{Y}_{in}$, and
3. $\text{conv } A[\bar{X}] + \text{cone } \mathcal{Y}_{in} \subseteq \mathcal{A}$.

Note the close resemblance between these properties and those in Definitions 3.6 and 3.7.

The set $\mathcal{Y}_{in}$ might, however, be empty for an unbounded problem (P), see Example 7.2. In such a case, the inner approximation $\text{conv } A[\bar{X}] + \text{cone } \mathcal{Y}_{in}$ is a bounded set despite $\mathcal{A}$ being unbounded. Although such inner approximation is of limited quality, the problem is nevertheless solved in the sense of Definition 3.6—as the diameter of the set $\mathcal{Y}_{out}$ is bounded by δ .

Let us now discuss the right-hand side inclusions of (6.5). The polyhedrons $\mathcal{A}_0$ and $\text{conv } A[\bar{X}] + \text{cone } \mathcal{Y}_{out} + B_\epsilon$ are both (ϵ, δ) -outer approximations of the image set $\mathcal{A}$, where out of the two outer approximations $\mathcal{A}_0$ is the better one. Therefore, if an approximation of the set rather than a solution is of interest, one should choose (depending on the application) between the inner approximation $\text{conv } A[\bar{X}] + \text{cone } \mathcal{Y}_{in}$ and the outer approximation $\mathcal{A}_0$.

Remark 6.8 Algorithms 6.1 and 6.5 assume that the optimal solutions of the scalarizations are found. As we discussed in Remarks 4.8 and 5.5, in practice only near-optimal solutions will be found. We again denote by ϵ the accuracy to which these near-optimal solutions are

found. It is reasonable to assume $\varepsilon \ll \min\{\epsilon, \delta\}$. Treating the found near-optimal solutions as optimal solutions (i.e. omitting ε within the relevant half-space) means that the half-space used to generate the outer approximation $\mathcal{A}_0$ are slightly shifted, see Remark 4.8. However, these half-spaces are not tilted, therefore the recession cone of $\mathcal{A}_0$ is not affected and we still approximate $\mathcal{A}_\infty$ correctly.

Remark 6.9 Note that correctness of Algorithms 6.1 and 6.5 was proven under the assumption that they terminate. Finite termination remains, similarly to Algorithm 5.1, an open question.

7 Examples

Example 7.1 We compute approximations of an intersection of two ellipses projected onto a lower dimensional space. Specifically, we approximate the two-dimensional set

$$\left\{ (x_1, x_2) \mid \exists x_3 \in \mathbb{R} : x_1^2 + \frac{(x_2 - 1)^2}{4} + x_3^2 \leq 1, \quad \frac{(x_1 - 1)^2}{4} + x_2^2 + \frac{(x_3 - 1)^2}{4} \leq 1 \right\} \quad (7.1)$$

and the three-dimensional set

$$\left\{ (x_1, x_2, x_3) \mid \exists x_4 \in \mathbb{R} : x_1^2 + \frac{(x_2 - 1)^2}{4} + x_3^2 + \frac{(x_4 - 1)^2}{4} \leq 1, \right. \\ \left. \frac{(x_1 - 1)^2}{4} + x_2^2 + \frac{(x_3 - 1)^2}{4} + x_4^2 \leq 1 \right\}. \quad (7.2)$$

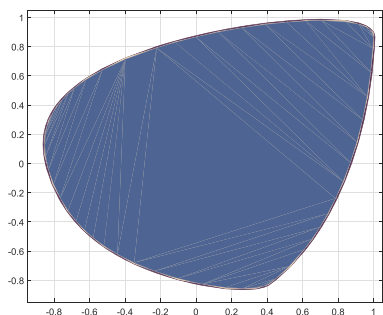
These convex projection problems (or, more precisely, their shifted versions) were solved via the associated multi-objective problems in [10]. Here we apply Algorithm 5.1. We compare the original version of the algorithm with the alternative initialization described in Remark 5.2, both for a tolerance $\epsilon = 0.01$. For the set (7.1), the alternative initialization leads to fewer computations needed (60 optimizations and 6 evaluations of a polyhedron for the original initialization versus 54 optimizations and 5 evaluations of a polyhedron for the alternative). However, for the set (7.2) the opposite is the case (1544 optimizations and 7 evaluations of a polyhedron for the original version and 1570 optimizations and 8 evaluations of a polyhedron for the alternative). Figure 1 visualizes the obtained approximations.

Three approximations for each set are obtained: the inner approximation $\text{conv } A[\bar{X}]$, the outer approximation $\mathcal{A}_0$ and the outer approximation $\text{conv } A[\bar{X}] + B_\epsilon$, see also the discussion around Equation (6.5). All three approximations are depicted in Fig. 1a, but they are not visually distinguishable. For the sake of visualization, Fig. 2 contains zoomed-in versions. There we can also observe that the outputs of the algorithm with different initializations differ.

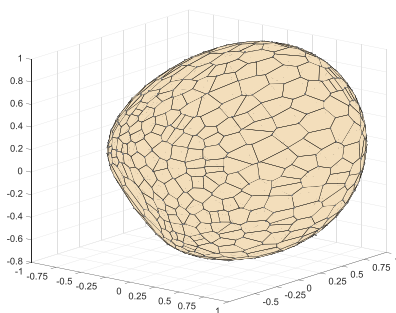
Example 7.2 Now we provide an illustrative example of an unbounded problem: We approximate the set

$$\{(x_1, x_2) : (x_1 \cos \theta + x_2 \sin \theta)^2 \leq (x_1 \sin \theta + x_2 \cos \theta)\} \quad (7.3)$$

for $\theta = 0$ and $\theta = \frac{\pi}{6}$. These are the quadratic epigraph and its rotated version. We provide Algorithm 6.5 with a known element of the set as an input. Note that while the theoretical results were derived only for input points from the relative interior of $A[\mathbb{X}]$, one can numerically also experiment with boundary input points. In this example, we input the point

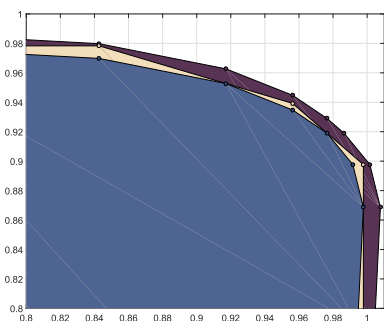


(a) Approximations of (7.1). At this scale the output of different initializations are not visually distinguishable.

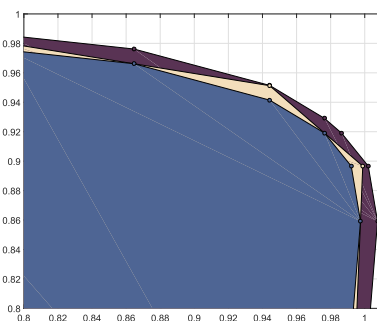


(b) Outer approximation $\mathcal{A}_0$ of (7.2).

Fig. 1 Approximations of (7.1) and (7.2) from Example 7.1 for tolerance $\epsilon = 0.01$



(a) Original initialization of Algorithm 5.1.



(b) Alternative initialization of Algorithm 5.1.

Fig. 2 Zoomed-in approximations of (7.1) from Example 7.1 for tolerance $\epsilon = 0.01$: Inner approximation $\text{conv } A[\bar{X}]$ in blue, outer approximation $\mathcal{A}_0$ in yellow and outer approximation $\text{conv } A[\bar{X}] + B_\epsilon$ in purple

$v = (0, 2)$ for $\theta = 0$ and $v = (\cos \theta - \sin \theta, \sin \theta + \cos \theta)$ for $\theta = \frac{\pi}{6}$. For both cases we run the algorithm for tolerance parameters set to $\delta = 0.1$ and $\epsilon = 0.01$.

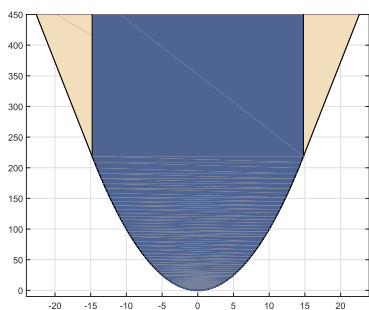
The obtained approximations for the case of $\theta = 0$ are illustrated in Fig. 3a. The algorithm computes a solution consisting of

$$\mathcal{Y}_{in} = \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}, \quad \mathcal{Y}_{out} = \left\{ \begin{pmatrix} -0.0327 \\ 0.9673 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0.0327 \\ 0.9673 \end{pmatrix} \right\}$$

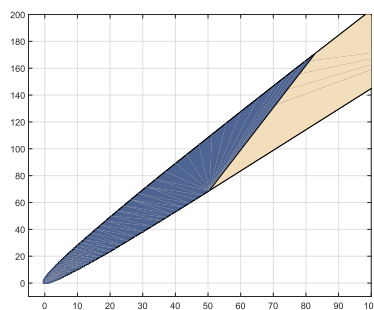
as well as 149 elements of $\bar{X}$. In this case an element of $\mathcal{Y}_{in}$ is found in the first run of Step 5 of Algorithm 6.1, since the recession direction coincides with a vertex of a unit ball.

This is not the case for the set rotated by $\theta = \frac{\pi}{6}$. No recession direction is found and Step 5 of Algorithm 6.1 terminates when the diameter of $\mathcal{Y}_{out}$ becomes sufficiently small. The algorithm provides a solution of the problem consisting of

$$\mathcal{Y}_{in} = \emptyset, \quad \mathcal{Y}_{out} = \left\{ \begin{pmatrix} 0.3789 \\ 0.6211 \end{pmatrix}, \begin{pmatrix} 0.3459 \\ 0.6541 \end{pmatrix} \right\}$$

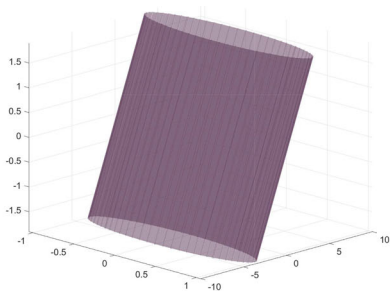


(a) Approximations of the set (7.3) for $\theta = 0$. The algorithm was provided with an inner point of the set and it performed 153 optimizations and 13 evaluations of a polyhedron.

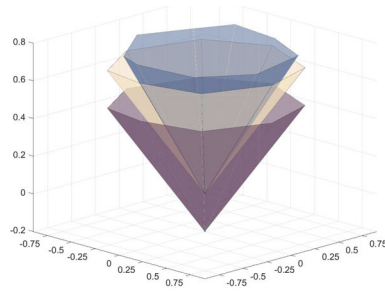


(b) Approximations of the set (7.3) for $\theta = \frac{\pi}{6}$. The algorithm was provided with a boundary point of the set and it performed 130 optimizations and 11 evaluations of a polyhedron.

Fig. 3 Approximations of (7.3) from Example 7.2: Inner approximation $\text{conv } A[\bar{X}] + \text{cone } \mathcal{Y}_{in}$ in blue and outer approximation $\mathcal{A}_0$ in yellow. The outer approximation $\text{conv } A[\bar{X}] + \text{cone } \mathcal{Y}_{out} + B_\epsilon$ is not visually distinguishable from $\mathcal{A}_0$



(a) Approximation of (7.4). The algorithm performed 71 optimizations and 7 evaluations of a polyhedron.



(b) Approximations of (7.5). The algorithm performed 31 optimizations and 8 evaluations of a polyhedron.

Fig. 4 Approximations of (7.4) and (7.5) from Examples 7.3 and 7.4

and 129 elements of $\bar{X}$. Note that this means that the inner approximation is bounded despite the set (and the problem) being unbounded. Approximations are displayed in Fig. 3b.

Example 7.3 We also provide an example of a set with a lineality space. We approximate the unbounded 'tube-like' set

$$\{(x_1, x_2, x_3) : x_1^2 + (x_2 \cos \theta - x_3 \sin \theta)^2 \leq 1\} \quad (7.4)$$

for $\theta = \frac{\pi}{3}$ via Algorithm 6.5 for tolerances $\delta = 0.1$ and $\epsilon = 0.01$. In this case we generate the point $v \in A[\bar{X}]$ through a feasibility check. Since the three approximations provided by the algorithm are not well visually distinguishable, we only display one of them in Fig. 4a.

Example 7.4 We revisit the example of approximating the ice cream cone

$$\{(x_1, x_2, x_3) : \sqrt{x_1^2 + x_2^2} \leq x_3\}, \quad (7.5)$$

which was solved in [18] via a convex vector optimization problem. This time we treat the problem as an instance of (P) and solve it via Algorithm 6.5 for the same tolerance $\delta = 0.2$ as in [18], which also allows to visually distinguish the three approximations. The approximations are displayed in Fig. 4b. Note that while the inner approximation $\text{conv } A[\tilde{X}] + \text{cone } \mathcal{Y}_{in}$ (in blue) and the outer approximation $\mathcal{A}_0$ (in yellow) are in this case cones, the second outer approximation $\text{conv } A[\tilde{X}] + \text{cone } \mathcal{Y}_{out} + B_\epsilon$ (in purple) is a shifted cone that does not start in the origin because of the presence of the (polyhedral) ball B_ϵ .

8 Conclusions

The existing literature provides algorithms and methods for the bounded case of a convex projection problem. These are based on (explicitly or implicitly) transforming the convex projection into an associated multi-objective convex problem with one additional dimension of an image space. Consequently, the existing methods perform operations in the higher-dimensional image space of the multi-objective problem.

The first contribution of this paper is Algorithm 5.1, which solves a bounded convex projection directly, without a transformation to associated multi-objective problem. Algorithm 5.1 is a Benson-type, outer approximation algorithm performing operations in the (lower dimensional) image space of the projection and is inspired by [1, Algorithm 1]. Just like for the outer approximation [1, Algorithm 1] or [12, Algorithm 1], we provide correctness result but finite termination remains an open question.

Algorithm 5.1 serves as a stepping stone to the main result of this paper, a method for solving unbounded convex projection. We introduce an appropriate solution concept and develop an algorithm computing such solution. Algorithm 6.1 computes (inner and outer) approximations of a recession cone of a convex projection; it builds on the ideas of [18, Algorithm 1] for convex vector optimization problems. Finally, Algorithm 6.5 combines the ideas of Algorithm 5.1 and Algorithm 6.1 to compute a solution of a convex projection problem. While correctness is proven, finite termination of these (or modified) algorithms remains a question for future research.

Acknowledgements The authors thank the anonymous referees for their valuable suggestions and questions. In particular, we thank one of the referees for bringing our attention to the example in Remark 4.2. G. Kováčová acknowledges support from the Society for Industrial and Applied Mathematics (SIAM) as part of the SIAM Postdoctoral Support Program, which is funded by contributions to the SIAM Postdoctoral Support Fund, established by a gift from Drs. Martin Golubitsky and Barbara Keyfitz.

Funding Open access funding provided by Vienna University of Economics and Business (WU).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Ararat, Ç., Ulus, F., Umer, M.: A norm minimization-based convex vector optimization algorithm. *J. Optim. Theory Appl.* **194**(2), 681–712 (2022)
2. Bronstein, E.M.: Approximation of convex sets by polytopes. *J. Math. Sci.* **153**(6), 727 (2008)
3. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press (2004)
4. Dörfler, D., Löhne, A.: Polyhedral approximation of spectrahedral shadows via homogenization. *J. Optim. Theory Appl.* **200**, 874–890 (2024)
5. Dörfler, D.: On the approximation of unbounded convex sets by polyhedra. *J. Optim. Theory Appl.* **194**, 265–287 (2022)
6. Feinstein, Z., Rudloff, B.: A recursive algorithm for multivariate risk measures and a set-valued Bellman's principle. *J. Global Optim.* **68**(1), 47–69 (2017)
7. Feinstein, Z., Rudloff, B., Zhang, J.: Dynamic set values for nonzero sum games with multiple equilibriums. *Math. Oper. Res.* **47**(1), 616–642 (2022)
8. Kamenev, G.K.: A class of adaptive algorithms for approximating convex bodies by polyhedra. *Comput. Math. Math. Phys.* **32**(1), 114–127 (1992)
9. Kováčová, G., Rudloff, B.: Time consistency of the mean-risk problem. *Oper. Res.* **69**(4), 1100–1117 (2021)
10. Kováčová, G., Rudloff, B.: Convex projection and convex multi-objective optimization. *J. Global Optim.* **83**, 301–327 (2022)
11. Kováčová, G., Rudloff, B., Cialenco, I.: Acceptability maximization. *Front. Math. Financ.* **1**(2), 219–248 (2022)
12. Löhne, A., Rudloff, B., Ulus, F.: Primal and dual approximation algorithms for convex vector optimization problems. *J. Global Optim.* **60**(4), 713–736 (2014)
13. Löhne, A., Weißing, B.: Equivalence between polyhedral projection, multiple objective linear programming and vector linear programming. *Math. Methods Oper. Res.* **84**(2), 411–426 (2016)
14. Löhne, A., Zhao, F., Shao, L.: On the approximation error for approximating convex bodies using multiobjective optimization. *Appl. Set-Valued Anal. Optim.* **3**, 341–354 (2021)
15. Ney, P., Robinson, S.: Polyhedral approximation of convex sets with an application to large deviation probability theory. *J. Convex Anal.* **2**(1), 229–240 (1995)
16. Shao, L., Zhao, F., Cong, Y.: Approximation of convex bodies by multiple objective optimization and an application in reachable sets. *Optimization* **67**(6), 783–796 (2018)
17. Ulus, F.: Tractability of convex vector optimization problems in the sense of polyhedral approximations. *J. Global Optim.* **72**(4), 731–742 (2018)
18. Wagner, A., Ulus, F., Rudloff, B., Kováčová, G., Hey, N.: Algorithms to solve unbounded convex vector optimization problems. *SIAM J. Optim.* **33**(4), 2598–2624 (2023)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.