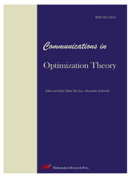
Commun. Optim. Theory 2026 (2026) 4 https://doi.org/10.23952/cot.2026.4

MATHRES

Communications in Optimization Theory

Available online at http://cot.mathres.org



AN ACCELERATION TECHNIQUE FOR METHODS FOR FINDING THE NEAREST POINT IN A POLYTOPE AND COMPUTING THE DISTANCE BETWEEN TWO POLYTOPES

MAXIM VLADIMIROVICH DOLGOPOLIK

Laboratory of Control of Complex Systems, Institute for Problems in Mechanical Engineering of the Russian Academy of Sciences, Bolshoy pr. V.O., 61, Saint Petersburg, 199178, Russia

Dedicated to the memory of Professor Hedy Attouch

Abstract. We present a simple and efficient acceleration technique for an arbitrary method for computing the Euclidean projection of a point onto a convex polytope, defined as the convex hull of a finite number of points, in the case when the number of points in the polytope is much greater than the dimension of the space. The technique consists in applying any given method to a "small" subpolytope of the original polytope and gradually shifting it, till the projection of the given point onto the subpolytope coincides with its projection onto the original polytope. The results of numerical experiments demonstrate the high efficiency of the proposed acceleration technique. In particular, they show that the reduction of computation time increases with an increase of the number of points in the polytope and is proportional to this number for some methods. In the second part of the paper, we also discuss a straightforward extension of the proposed acceleration technique to the case of arbitrary methods for computing the distance between two convex polytopes, defined as the convex hulls of finite sets of points.

Keywords. Acceleration; Convex hull; Distance computation; Minimal norm; Polytope.

2020 Mathematics Subject Classification. 65D18, 65K10.

1. Introduction

The problems of computing the Euclidean projection of a point onto a polytope and the distance between two polytopes are one of central problems of computational geometry whose importance for applications cannot be overstated. A need for fast and reliable methods for solving these problems arises in nonsmooth optimization [1, 2], submodular optimization [4, 8, 9], support vector machine algorithms [3, 18, 23, 34], and many other applications.

Since the mid 1960s, a vast array of methods for finding the nearest point in a polytope and computing the distance between two polytopes has been developed. In the case of the nearest point problem, among them are various methods based on quadratic programming (e.g. the Gilbert method [13]), the Wolfe method [32] (see also the recent complexity analysis of

E-mail address: dmv@ipme.ru

Received September 27, 2024; Accepted March 4, 2025.

a version of this method [5]), the Mitchell-Demyanov-Malozemov (MDM) method [24] and its modifications [3, 34], subgradient algorithms based on nonsmooth penalty functions [31], geometric algorithms [25, 30], a dual algorithm [10], etc. Let us also mention that a number of nontrivial equivalent reformulations of the nearest point problem was discussed in [12].

Although the problem of computing the distance between two polytopes can be easily reduced to the nearest point problem and the aforementioned methods can be applied to find its solution, several specialized methods for solving the distance problem have been developed as well. A highly efficient method for computing the distance between two convex hulls in three-dimensional space was developed by Gilbert et al. [14], while methods for computing the distance in the two-dimensional case were studied in [6, 7, 15, 33]. A modification of the MDM method for computing the distance between two convex hulls in space of arbitrary dimension was proposed [16, 17], while an algorithm for solving this problem based on the fixed-point theory was studied in [22]. Let us also mention a dual algorithm [11], a recursive algorithm [29], the sequential minimal optimization method [27], and some less known methods [19, 28] for computing the distance between two polytopes.

Our main goal is to develop and analyse a general acceleration technique that can be applied to *any* method for computing the Euclidean projection of a point onto a polytope, defined as the convex hull of a finite number of points, in the case when the number of points is significantly greater than the the dimension of the space. We present this technique as a meta-algorithm that on each iteration employs a chosen algorithm for finding the nearest point in a polytope as a subroutine. The acceleration technique itself consists in applying the given algorithm to a "small" subpolytope of the original polytope and gradually shifting it with each iteration till the required projection is computed.

We study both a theoretical version of the proposed acceleration technique and its robust version that is more suitable for practical implementation, since it takes into account finite precision of computations. We prove correctness and finite termination of both these versions under suitable assumptions, and present some very promising results of numerical experiments. These results demonstrate a drastic reduction of computation time for several different methods for finding the nearest point in a polytope achieved with the use of our acceleration technique. Furthermore, our numerical experiments showed that the reduction of computation time increases with an increase of the number of points ℓ in the polytope and is proportional to this number for large ℓ.

It should be noted that the proposed acceleration technique shares many similarites with the Wolfe method [32] and the Frank-Wolfe algorithms [20, 21]. Nonetheless, there are important differences between these methods related to the way in which they remove redundant points on each iteration. We present a theoretical discussion of these differences and some results of numerical experiments demonstrating how a different way of removing redundant points presented in this paper leads to a significant reduction of the number of iterations (shifts of the subpolytope) in comparison with the Wolfe method.

In the second part of the paper, we extend the proposed acceleration technique to the case of *arbitrary* methods for computing the Euclidean distance between two polytopes, defined as the convex hulls of finite sets of points, in the case when the number of points in each of these sets is significantly greater than the dimension of the space. We present a theoretical analysis of this extension and some results of numerical experiments demonstrating the high efficiency of the acceleration technique in the case of the distance problem.

The paper is organised as follows. Section 2 contains a detailed analysis and discussion of an acceleration technique for arbitrary methods for computing the Euclidean projection of a point onto a polytope. Subsections 2.1 and 2.2 are devoted to the study of a theoretical scheme of the acceleration technique, while Subsection 2.3 contains a robust version of this technique that is more suitable for practical implementation. Some promising results of numerical experiments for the proposed acceleration technique are collected in Subsection 2.4, while Section 3 contains a discussion of the differences between our acceleration technique and the Wolfe method, as well as some results of numerical experiments highlighting these differences. Finally, a straighforward extension of this acceleration technique to the case of method for computing the distance between two polytopes is studied in Section 4.

2. Finding the Nearest Point in a Polytope

In this section, we study a general acceleration technique for an *arbitrary* algorithm for computing the Euclidean projection of a point onto a polytope, defined as the convex hull of a finite number of points in $\mathbb{R}^d$, in the case when the number of points in the polytope is much greater than the dimension of the space. In other words, we discuss an acceleration technique for methods of solving the following optimization problem

$$\min_x \|x - z\| \quad \text{subject to} \quad x \in P := \operatorname{co}\{x_1, \ldots, x_\ell\} \subset \mathbb{R}^d \tag{$\mathscr{P}$}$$

in the case when $\ell \gg d$. Here $z, x_1, \ldots, x_\ell \in \mathbb{R}^d$ are given points and $\|\cdot\|$ is the Euclidean norm.

2.1. General acceleration technique.

Before we proceed to the description of the acceleration technique, let us first recall the following well-known optimality condition for the problem $(\mathscr{P})$ (see, e.g. [32, 24]), for the sake of completeness.

Proposition 2.1. *A point $x_* \in P$ is a globally optimal solution of the problem $(\mathscr{P})$ if and only if*

$$\langle x_* - z, x_i - x_* \rangle \geq 0 \quad \forall i \in I = \{1, \ldots, \ell\}. \tag{2.1}$$

Suppose that an algorithm $\mathscr{A}$ for solving the nearest point problem $(\mathscr{P})$ is given. For any point $w \in \mathbb{R}^d$ and any polytope $Q \subset \mathbb{R}^d$ the algorithm returns a unique solution $x_* = \mathscr{A}(w, Q)$ of the problem

$$\min_{x \in Q} \|x - w\|.$$

No other assumptions on the algorithm $\mathscr{A}$ are imposed.

Remark 2.2. It should be noted that throughout this article we implicitly view all algorithms not in the way they are viewed in computer science and the theory of algorithms, but simply as single-valued maps. In particular, the algorithm $\mathscr{A}$ is a single-valued function mapping the Cartesian product of $\mathbb{R}^d$ and the set of all polytopes $Q \subset \mathbb{R}^d$ into the space $\mathbb{R}^d$. It can be explicitly defined as $\mathscr{A}(w, Q) = \arg\min_{x \in Q} \|x - w\|$. One can also look at algorithm $\mathscr{A}$ as a black box with input (w, Q) and output $\arg\min_{x \in Q} \|x - w\|$, which is akin to the way oracles are viewed within optimization theory (cf. [26]).

Our aim is to design an acceleration technique for the algorithm $\mathscr{A}$ that would improve its performance. This acceleration technique will be presented as a meta-algorithm that utilises the algorithm $\mathscr{A}$ as a subroutine on each iteration.

Recall that we are only interested in the case when the number of points ℓ is significantly greater than the dimension of the space d. Furthermore, nothing is known about the structure of the algorithm $\mathscr{A}$. Therefore, perhaps, the only straightforward way to potentially accelerate this algorithm is by applying $\mathscr{A}$ to a polytope generated by a relatively small number of points $\{x_{i_1}, \ldots, x_{i_s}\}$ from the set $\{x_1, \ldots, x_\ell\}$ and then replacing some of the extreme points of the polytope $P_0 = \operatorname{co}\{x_{i_1}, \ldots, x_{i_s}\}$ with different points from P and repeating the same procedure till the projection of z onto P_0 coincides with the projection of z onto P. The key ingredient of this strategy is *an exchange rule* that replaces points from P_0 by different points from P.

Denote $\mathbb{N} = \{0, 1, 2, \ldots\}$ and $\mathbb{N}^* = \mathbb{N} \setminus \{0\}$. To define an accelerating meta-algorithm for solving the problem $(\mathscr{P})$, we need to choose parameter $s \in \mathbb{N}^*$ that defines the size of the *subpolytope* P_0 and an exchange rule $\mathscr{E}$ whose output consists of two index sets I_1 and I_2. The first set $I_1 \subseteq \{i_1, \ldots, i_s\}$ defines the points in P_0 that must be removed from P_0, while the second index set $I_2 \subseteq I \setminus \{i_1, \ldots, i_s\}$ defines the points that are included into P_0 before the next iteration. In the general case, the index sets I_1 and I_2 might have arbitrary sizes, that change with each iteration, and I_1 can be empty. However, for the sake of simplicity, we restrict our consideration to the case when the cardinalities $|I_1|$ and $|I_2|$ of the sets I_1 and I_2 coincide and are equal to some number $q \in \mathbb{N}^*$. Any exchange rule $\mathscr{E}$ that for a given input consisting of s indices $\{i_1, \ldots, i_s\}$ returns two collections of indices (I_1, I_2) with

$$I_1 \subseteq \{i_1, \ldots, i_s\}, \quad I_2 \subseteq I \setminus \{i_1, \ldots, i_s\}, \quad |I_1| = |I_2| = q$$

is called an (s,q)-*exchange rule*.

Remark 2.3. It should be noted that an actual exchange rule obviously requires more input parameters than just a set of indices $\{i_1, \ldots, i_s\}$. In particular, it might need some information about the point z, the set $\{x_1, \ldots, x_\ell\}$, the nearest point $\mathscr{A}(z, P_0)$ in the polytope P_0, etc. as its input. However, for the sake of shortness, below we explicitly indicate only a set of indices as an input of an exchange rule $\mathscr{E}$.

Thus, we arrive at the following scheme of the meta-algorithm for solving the problem $(\mathscr{P})$ given in Meta-algorithm 1. In its core, this meta-algorithm consists in choosing a *subpolytope* P_n of the original polytope P and finding the projection y_n of the given point z onto P_n. If this projection happens to coincide with the projection of z onto P (this fact is verified with the use of optimality conditions from Proposition 2.1), then the meta-algorithm terminates. Otherwise, one replaces some points in P_n, thus constructing a new subpolytope P_{n+1}, and repeats the same procedure till the projection of z onto P_n coincides with the projection of z onto the original polytope P.

From the geometric point of view, the meta-algorithm consists in choosing a "small" subpolytope P_n in the original polytope P and gradually "shifting" it with each iteration till P_n contains the projection of z onto P. This procedure is performed with the hope that it is much faster to compute a projection of a point onto a small subpolytope and gradually shift it, rather than to compute the projection of this point onto the original polytope with a very large number of vertices (i.e. with $\ell \gg d$). The results of numerical experiments reported below demonstrate that this hope is fully justified.

Algorithm 1 Meta-algorithm for finding the nearest point in a polytope.

```
**Input:** a point $z \in \mathbb{R}^d$, a collection of points $\{x_1, \ldots, x_\ell\} \subset \mathbb{R}^d$, an algorithm $\mathscr{A}$ for solving the nearest point problem, parameters $s, q \in \{1, \ldots, \ell\}$ with $s \geq q$, and an $(s,q)$-exchange rule $\mathscr{E}$.
**Initialization:** Put $n = 0$, choose an index set $I_n \subseteq I$ with $|I_n| = s$, and define a polytope $P_n = \operatorname{co}\{x_i \mid i \in I_n\}$.
**Step 1:** Compute $y_n = \mathscr{A}(z, P_n)$. If $y_n$ satisfies the optimality condition
    $\langle y_n - z, x_i - y_n \rangle \geq 0 \quad \forall i \in I,$    (2.2)
**return** $y_n$.
**Step 2:** Compute $(I_{1n}, I_{2n}) = \mathscr{E}(I_n)$ and define
    $I_{n+1} = \Big(I_n \setminus I_{1n}\Big) \cup I_{2n}, \quad P_{n+1} = \operatorname{co}\big\{x_i \bigm| i \in I_{n+1}\big\}.$
Set $n = n + 1$ and go to **Step 1**.
```

The two following lemmas describe simple conditions on the exchange rule $\mathscr{E}$ ensuring that the proposed meta-algorithm indeed solves the problem $(\mathscr{P})$ and, furthermore, terminates after a finite number of iterations. These lemmas provide one with two convenient criteria for choosing effective exchange rules. Although (rather awkward) proofs of these lemmas are obvious, we present them for the sake of completeness.

For any set $\Omega \subset \mathbb{R}^d$ and any $x \in \mathbb{R}^d$ denote by $\operatorname{dist}(x, \Omega) = \inf_{y \in \Omega} \|x - y\|$ the distance from x to Ω.

Lemma 2.4. *Suppose that the exchange rule $\mathscr{E}$ satisfies* ***the distance decay condition****: if for some $n \in \mathbb{N}$ the point y_n does not satisfy optimality condition* (2.2)*, then*

$$\operatorname{dist}(z, P_{n+1}) < \operatorname{dist}(z, P_n). \tag{2.3}$$

Then Meta-algorithm 1 terminates after a finite number of steps and returns an optimal solution of the problem $(\mathscr{P})$.

Proof. From Proposition 2.1 and the termination criterion (2.2) of Meta-algorithm 1 it follows that if this algorithm terminates in a finite number of steps, then the last computed point y_n (and the output of Meta-algorithm 1) is an optimal solution of the problem $(\mathscr{P})$. Thus, we only need to prove that the algorithm terminates in a finite number of steps.

To prove the finite termination, note that there is only a finite number of distinct subsets of the set $\{x_1, \ldots, x_\ell\}$ with cardinality s. Moreover, from the distance decay condition (2.3) it follows that if the algorithm does not terminate in $n \in \mathbb{N}$ iterations, then all index sets $I_0, I_1, \ldots, I_n$ are distinct. Therefore, in a finite number of steps the algorithm must find a point y_n satisfying the termination criterion (2.2). □

Remark 2.5. If the exchange rule $\mathscr{E}$ satisfies the distance decay condition, then Meta-algorithm 1 generates a finite sequence of polytopes $\{P_n\} \subset P$ such that $P_n \neq P_k$ for any $n \neq k$. Note that the length of such sequence does not exceed the number of s-combinations of the set $\{1, \ldots, \ell\}$, which is equal to $\binom{\ell}{s} = \mathscr{O}(\ell^s)$. Therefore, Meta-algorithm 1 has polynomial in ℓ complexity, provided the exchange rule $\mathscr{E}$ satisfies the distance decay condition and has polynomial in ℓ complexity as well.

Lemma 2.6. *Let $\ell \geq d+1$ and an (s,q)-exchange rule $\mathscr{E}$ with $s \geq q$ satisfy the distance decay condition for any point $z \in \mathbb{R}^d$ and any polytope $P = \operatorname{co}\{x_1, \ldots, x_\ell\}$. Then $s \geq d+1$.*

Proof. Suppose by contradiction that $s \leq d$. Let us provide a particular point z and a particular polytope P for which any (s,q)-exchange rule with $q \leq s \leq d$fails to satisfy the distance decay condition.

Let $z = 0$. If $d = 1$, define $\ell = 2$, $x_1 = 1$, and $x_2 = -1$. In this case $s = q = 1$ and for any choice of I_0 it is obviously impossible to satisfy the condition $\operatorname{dist}(z, P_1) < \operatorname{dist}(z, P_2)$.

Let now $d \geq 2$. Put $z = 0$, $\ell = d+1$, and define the points x_i as follows:

$$x_1 = (1,0,\ldots,0,-1), x_2 = (0,1,0,\ldots,0,-1), \ldots, x_{d-2} = (0,\ldots,0,1,0,-1),$$
$$x_{d-1} = (-1,\ldots,-1,1,-1), \quad x_d = (-1,\ldots,-1), \quad x_{d+1} = (0,\ldots,0,1).$$

As is easily seen, any d points from the set $\{x_1, \ldots, x_{d+1}\}$ are linearly independent and, in addition, $0 \in P$, since

$$\sum_{i=1}^{d+1} \alpha_i x_i = 0, \quad \sum_{i=1}^{d+1} \alpha_i = 1$$

for

$$\alpha_1 = \ldots = \alpha_{d-2} = \frac{1}{2(d-1)}, \quad \alpha_{d-1} = \alpha_d = \frac{1}{4(d-1)}, \quad \alpha_{d+1} = \frac{1}{2}.$$

Hence, in particular, for any index set $K \subset I$ with $|K| = s \leq d$ one has $\operatorname{dist}(z, P_K) > 0$, where $P_K = \operatorname{co}\{x_i \mid i \in K\}$. Therefore, for any choice of an (s,q)-exchange rule with $s \leq d$ and $s \geq q$ the stopping criterion (2.2) cannot be satisfied, which by the previous lemma implies that this exchange rule does not satisfy the distance decay condition. □

Remark 2.7. One can readily verify that if in the lemma above one imposes the additional assumption that $z \notin P$, then the statement of the lemma holds true for $s \geq d$. To prove this result, one simply needs to put $z = 0$, $\ell = d$, and define P as the convex hull of the first d points from the proof of the lemma above.

2.2. **The steepest descent exchange rule.**

Let us present a detailed analysis of a particular exchange rule based on the optimality condition from Proposition 2.1 (or, equivalently, the stopping criterion (2.2)) and satisfying the distance decay condition for any point z and any polytope P.

Bearing in mind Lemma 2.6, we propose to consider the following $(d+1,1)$-exchange rule that on each iteration of Meta-algorithm 1 replaces only one point in the current polytope P_n. Suppose that for some $n \in \mathbb{N}$ the stopping criterion (2.2) is not satisfied. Then we define the new point x_{2n} from P, that is included into P_{n+1}, as any point from P on which the minimum in

$$\min_{i \in I} \langle y_n - z, x_i - y_n \rangle$$

is attained (cf. a similar rule for including new points in the major cycle of the Wolfe method [32]).

To find a point that is removed from P_n, note that $z \notin P_n$ due to the definition of y_n and the fact that condition (2.2) does not hold true. Therefore, y_n belongs to the boundary of P_n, which by [35, Lemma 2.8] implies that y_n is contained in a face of P_n of dimension at most $d-1$. Hence by [35, Propositions 1.15 and 2.3] the point y_n can be represented as a convex combination of at most d points from the set $\{x_i \mid i \in I_n\}$. In other words, there exists $i_{1n} \in I_n$ such that $y_n \in \operatorname{co}\{x_i \mid I_n \setminus \{i_{1n}\}\}$, and it is natural to remove the point $x_{i_{1n}}$ from the polytope P_n.

Thus, we arrive at the following theoretical scheme of a $(d+1,1)$-exchange rule that we call *the steepest descent exchange rule*:

- **Input:** an index set $I_n \subset I$ with $|I_n| = d+1$, the point z, the set $\{x_1, \ldots, x_\ell\}$, and the projection y_n of z onto $P_n = \operatorname{co}\{x_i \mid i \in I_n\}$.
- **Step 1:** Find $i_{1n} \in I_n$ such that $y_n \in \operatorname{co}\{x_i \mid I_n \setminus \{i_{1n}\}\}$.
- **Step 2:** Find $i_{2n} \in I$ such that

$$\langle y_n - z, x_{i_{2n}} \rangle = \min_{i \in I} \langle y_n - z, x_i \rangle.$$

Return $(\{i_{1n}\}, \{i_{2n}\})$.

From the discussion above it follows that the steepest descent exchange rule is correctly defined, provided y_n does not satisfy the stopping criterion (2.2). Let us verify that the steepest descent exchange rule always satisfies the distance decay condition. The proof of this result is almost the same as the proof of the analogous property for the iterates of the Wolfe method [32, 4]. We include a full proof of this result for the sake of completeness and due to the fact that Meta-algorithm 1 with the steepest descent exchange rule, strictly speaking, does not coincide with the Wolfe method (see Section 3 for more details).

Theorem 2.8. *For any point $z \in \mathbb{R}^d$ and any polytope $P = \operatorname{co}\{x_1, \ldots, x_\ell\}$ with $\ell \geq d+1$ the steepest descent exchange rule satisfies the distance decay condition.*

Proof. Suppose that for some $n \in \mathbb{N}$ the stopping criterion (2.2) does not hold true. Denote by

$$P_n^0 = \operatorname{co}\{x_i \mid i \in I_n \setminus \{i_{1n}\}\}$$

the polytope obtained from P_n after removing the point selected by the steepest descent exchange rule. By construction $y_n \in P_n^0$. Moreover, due to the definition of the exchange rule and the fact that the stopping criterion is not satisfied one has

$$\langle y_n - z, x_{i_{2n}} - y_n \rangle < 0. \tag{2.4}$$

Define $x_n(t) = (1-t)y_n + t x_{i_{2n}}$. Clearly, $x_n(t) \in P_{n+1}$ for any $t \in [0,1]$, since $P_{n+1} = \operatorname{co}\{P_n^0, x_{i_{2n}}\}$. Moreover, for any $t \in \mathbb{R}$ one has

$$f(t) := \|x_n(t) - z\|^2 = (1-t)^2 \|y_n - z\|^2 + 2t(1-t)\langle y_n - z, x_{i_{2n}} - z \rangle + t^2 \|x_{i_{2n}} - z\|^2.$$

Note that $f(0) = \|y_n - z\|^2$ and

$$f'(0) = -2\|y_n - z\|^2 + 2\langle y_n - z, x_{i_{2n}} - z \rangle = 2\langle y_n - z, x_{i_{2n}} - y_n \rangle < 0$$

due to (2.4). Therefore, for any sufficiently small $t \in (0,1)$ one has $f(t) < f(0)$, which implies that

$$\operatorname{dist}(z, P_{n+1}) = \min_{x \in P_{n+1}} \|z - x\| \leq \|z - x_n(t)\| = \sqrt{f(t)} < \sqrt{f(0)} = \|z - y_n\| = \operatorname{dist}(z, P_n).$$

Thus, the steepest descent exchange rule satisfies the distance decay condition for any point $z \in \mathbb{R}^d$ and any polytope $P = \operatorname{co}\{x_1, \ldots, x_\ell\}$ with $\ell \geq d+1$. □

Corollary 2.9. *Let $\ell \geq d+1$. Then Meta-algorithm 1 with $s = d+1$, $q = 1$, and the steepest descent exchange rule terminates after a finite number of iterations and returns an optimal solution of the problem $(\mathscr{P})$.*

Let us discuss a possible implementation of the steepest descent exchange rule. Clearly, the challenging part of this rule consists in finding an index $i_{1n} \in I_n$ such that $y_n \in \operatorname{co}\{x_i \mid I_n \setminus \{i_{1n}\}\}$. This difficulty can be overcome in the following way.

Namely, suppose that instead of returning the projection $u_* = \mathscr{A}(w, Q)$ of a point w onto a polytope $Q = \operatorname{co}\{u_1, \ldots, u_m\}$, the algorithm $\mathscr{A}$ actually returns a vector $\alpha = (\alpha^{(1)}, \ldots, \alpha^{(m)}) \in \mathbb{R}^m$ such that

$$u_* = \sum_{i=1}^m \alpha^{(i)} u_i, \quad \sum_{i=1}^m \alpha^{(i)} = 1, \quad \alpha_i \geq 0 \quad \forall i \in \{1, \ldots, m\}. \tag{2.5}$$

Let us note that for the vast majority of existing methods for finding the nearest point in a polytope this assumption either holds true by default or can be satisfied by slightly modifying the corresponding method (see [32, 24, 3, 34, 31, 25, 30]).

Let $\alpha_n = \mathscr{A}(z, P_n)$ and suppose that $\alpha_n^{(k)} = 0$ for some $k \in I_n$. Then one can obviously set $i_{1n} = k$ on Step 1 of the steepest descent exchange rule. To simplify the notation, hereinafter we identify the vector $\alpha_n = \mathscr{A}(z, P_n) \in \mathbb{R}^{d+1}$ with the extended vector $\widehat{\alpha}_n \in \mathbb{R}^\ell$ such that $\widehat{\alpha}_n^{(i)} = \alpha_n^{(i)}$ for any $i \in I_n$, and $\widehat{\alpha}_n^{(i)} = 0$ otherwise.

It should be noted that in the case when the points x_i, $i \in I_n$, are affinely independent and the stopping criterion (2.2) is not satisfied, there always exists $k \in I_n$ such that $\alpha_n^{(k)} = 0$. Indeed, as was noted above, in this case y_n does not belong to the interior of P_n, which by the characterization of interior points of a polytope [35, Lemma 2.8] implies that y_n *cannot* be represented in the form

$$y_n = \sum_{i \in I_n} \alpha^{(i)} x_i, \quad \sum_{i \in I_n} \alpha^{(i)} = 1, \quad \alpha^{(i)} > 0 \quad \forall i \in I_n.$$

Consequently, at least one of the coordinates of the vector $\alpha_n = \mathscr{A}(z, P_n)$ is equal to zero.

Even when the points x_i, $i \in I_n$, are affinely dependent, some methods (such as the the Wolfe method [32]) necessarily return a vector $\alpha_n = \mathscr{A}(z, P_n)$ with $\alpha_n^{(k)} = 0$ for at least one $k \in I_n$. However, other methods might return a vector α_n such that $\alpha_n^{(i)} > 0$ for all $i \in I_n$. In this case one can apply the following simple procedure, inspired the the proof of Carathéodory's theorem, to find the required index i_{1n}. This procedure is described in the algorithm below and we call it *the index removal method*:

- **Input:** an index set $I_n \subset I$ with $|I_n| = d + 1$, the point z, the set $\{x_1, \ldots, x_\ell\}$, and the vector $\alpha_n = \mathscr{A}(z, P_n)$.
- **Step 1:** Compute $\alpha_{\min} = \min_{i \in I_n} \alpha_n^{(i)}$. If $\alpha_{\min} = 0$, find $k \in I_n$ such that $\alpha_n^{(k)} = 0$ and **return** k.
- **Step 2:** Choose any $j \in I_n$, compute a least-squares solution γ_n of the system

$$\sum_{i \in I_n \setminus \{j\}} \gamma^{(i)} (x_i - x_j) = 0, \qquad \sum_{i \in I_n \setminus \{j\}} \gamma^{(i)} = 1, \tag{2.6}$$

  and set $\gamma_n^{(j)} = -1$. Find an index $k \in I_n$ on which the minimum in

$$\min \left\{ -\frac{\alpha_n^{(k)}}{\gamma_n^{(k)}} \;\middle|\; k \in I_n \colon \gamma_n^{(k)} < 0 \right\} \tag{2.7}$$

  is attained and **return** k.

The following proposition proves the correctness of the proposed method.

Proposition 2.10. *Suppose that for some $n \in \mathbb{N}$ the stopping criterion* (2.2) *does not hold true, and let $k \in I_n$ be the output of the index removal method. Then $y_n \in \operatorname{co}\{x_i \mid i \in I_n \setminus \{k\}\}$.*

Proof. The validity of the proposition in the case $\alpha_{\min} = 0$ is obvious. Therefore, let us consider the case $\alpha_{\min} > 0$, that is, the case when the index removal method executes Step 2. As was pointed out above, in this case the points x_i, $i \in I_n$, are necessarily affinely dependent, which by definition implies that the vectors $x_i - x_j$, $i \in I_n \setminus \{j\}$, are linearly dependent. Therefore system of linear equations (2.6) is consistent (despite being overdetermined) and its least-squares solution γ_n satisfies equations (2.6).

By definition $\gamma_n \neq 0$ and

$$\sum_{i \in I_n} \gamma_n^{(i)} x_i = 0, \quad \sum_{i \in I_n} \gamma_n^{(i)} = 0. \tag{2.8}$$

Moreover, $\gamma_n^{(j)} = -1$. Consequently, the minimum in (2.7), which we denote by λ, is correctly defined and $\lambda > 0$ (recall that $\alpha_{\min} > 0$).

Observe that

$$y_n = \sum_{i \in I_n} \alpha_n^{(i)} x_i = \sum_{i \in I_n} \alpha_n^{(i)} x_i + \lambda \sum_{i \in I_n} \gamma_n^{(i)} x_i = \sum_{i \in I_n} \xi_n^{(i)} x_i,$$

where $\xi_n^{(i)} = \alpha_n^{(i)} + \lambda \gamma_n^{(i)}$, by the first equality in (2.8). By the second equality in (2.8) and the definition of α_n (see (2.5)) one has $\sum_{i \in I_n} \xi_n^{(i)} = 1$. Moreover, if $\gamma_n^{(i)} \geq 0$, then clearly $\xi_n^{(i)} > 0$, while if $\gamma_n^{(i)} < 0$, then $\xi_n^{(i)} \geq 0$ by the definition of λ (see (2.7)). In addition, $\xi_n^{(k)} = 0$ by the definition of k, which obviously implies that $y_n \in \operatorname{co}\{x_i \mid i \in I_n \setminus \{k\}\}$. □

Remark 2.11. Let us point out that the second equation in (2.7) can obviously be replaced by the equation $\sum_{i \in I_n \setminus \{j\}} \gamma^{(i)} = C$ for any $C > 0$ (and one also has to set $\gamma_n^{(j)} = -C$).

Remark 2.12. It should be noted that in the general case the steepest descent exchange rule does not preserve the affine independence of the vectors x_i, $i \in I_n$, as the following simple example demonstrates. Let $d = 2$, $\ell = 4$, $z = 0$, and

$$x_1 = (2,2), \quad x_2 = (3,1), \quad x_3 = (1,1), \quad x_4 = (-1,1).$$

Put $I_0 = \{1,2,3\}$. Then $y_0 = x_3$, $\alpha_0 = \mathscr{A}(z, P_0) = (0,0,1)$, and the stopping criterion (2.2) is not satisfied. One can set $i_{10} = 1$, while by definition $i_{20} = 4$. Thus, $I_1 = \{2,3,4\}$. The points x_i, $i \in I_1$, are obviously affinely dependent, while the points x_i, $i \in I_0$, are affinely independent. Thus, the difficulty of finding the required index $i_{1n} \in I_n$ in the case when $\alpha_n^{(i)} > 0$ for all $i \in I_n$ (i.e. when the points x_i, $i \in I_n$, are affinely dependent) cannot be resolved by simply choosing an initial guess I_0 in such a way that the points x_i, $i \in I_0$, are affinely independent.

2.3. A robust version of the meta-algorithm.

It is clear that any algorithm $\mathscr{A}$ for solving the problem $(\mathscr{P})$ can find an optimal solution of this problem only in theory, while in practice it always returns an approximate solution of this problem due to finite precision of computations. Therefore it is an important practical issue to analyse the performance of Meta-algorithm 1 in the case when only computations with finite precision are possible.

Assume that instead of the "ideal" algorithm $\mathscr{A}$ one uses its "approximate" version $\mathscr{A}_\varepsilon$, $\varepsilon > 0$. The algorithm $\mathscr{A}_\varepsilon$ returns an approximate, in some sense (that will be specified below), solution $y_n(\varepsilon)$ of the nearest point problem. Then it is obvious that the stopping criterion

$$\langle y_n(\varepsilon) - z, x_i - y_n(\varepsilon) \rangle \geq 0 \quad \forall i \in I$$

of Meta-algorithm 1 cannot be satisfied and must be replaced by the inequality

$$\langle y_n(\varepsilon) - z, x_i - y_n(\varepsilon) \rangle \ge -\eta \quad \forall i \in I$$

for some small $\eta > 0$. The following well-known result (cf. [24]) indicates a direct connection between the inequality above and approximate optimality of $y_n(\varepsilon)$. For the sake of completeness, we include a full proof of this result.

Denote $\operatorname{diam}(P) = \max_{i,j \in I} \|x_i - x_j\|$. Observe that the inequality $\|x - y\| \le \operatorname{diam}(P)$ holds true for all $x, y \in P$, which means that $\operatorname{diam}(P)$ is indeed the diameter of the polytope P.

Proposition 2.13. *Let $y \in P$ satisfy the inequality*

$$\langle y - z, x_i - y \rangle \ge -\eta \quad \forall i \in I \tag{2.9}$$

for some $\eta > 0$. Then $\|y - x_*\| \le \sqrt{\eta}$, where x_* is an optimal solution of the problem $(\mathscr{P})$.

Conversely, let $y \in P$ be such that $\|y - x_*\| \le \varepsilon$ for some $\varepsilon > 0$. Then y satisfies inequality (2.9) *for any $\eta \ge (\operatorname{diam}(P) + \operatorname{dist}(z, P))\varepsilon$.*

Proof. Let a point $y \in P$ satisfy inequality (2.9) for some $\eta > 0$. Adding and subtracting z one gets

$$\|y - x_*\|^2 = \langle y - x_*, y - x_* \rangle = \langle y - z, y - x_* \rangle - \langle x_* - z, y - x_* \rangle.$$

Hence applying Proposition 2.1 one obtains

$$\|y - x_*\|^2 \le \langle y - z, y - x_* \rangle.$$

Since $x_* \in P$, there exist $\alpha_i \ge 0$, $i \in I$, such that $x_* = \sum_{i \in I} \alpha_i x_i$ and $\sum_{i \in I} \alpha_i = 1$. Therefore, with the use of (2.9) one finally gets that

$$\|y - x_*\|^2 \le \sum_{i=1}^{\ell} \alpha_i \langle y - z, y - x_i \rangle \le \sum_{i=1}^{\ell} \alpha_i \eta = \eta,$$

which implies that $\|y - x_*\| \le \sqrt{\eta}$.

Suppose now that $\|y - x_*\| \le \varepsilon$ for some $\varepsilon > 0$. Adding and subtracting x_* one has

$$\begin{aligned} \langle y - z, x_i - y \rangle &= \langle x_* - z, x_i - y \rangle + \langle y - x_*, x_i - y \rangle \\ &= \langle x_* - z, x_i - x_* \rangle + \langle x_* - z, x_* - y \rangle + \langle y - x_*, x_i - y \rangle. \end{aligned}$$

Hence with the use of Proposition 2.1 and the definition of x_* one gets that

$$\langle y - z, x_i - y \rangle \ge -\Big(\|x_* - z\| + \|x_i - y\| \Big) \|x_* - y\| \ge -\Big(\operatorname{dist}(z, P) + \operatorname{diam}(P) \Big) \varepsilon,$$

which implies the required result. □

Although a robust version of Meta-algorithm 1 can be formulated for an arbitrary exchange rule, for the sake of brevity we will formulate it only in the case of the steepest descent exchange rule. To this end, we suppose that the output of the approximate algorithm $\mathscr{A}_\varepsilon(z, P_n)$ is not an approximate solution $y_n(\varepsilon)$ of the nearest point problem $\min_{x \in P_n} \|x - z\|$, but a vector $\alpha_n(\varepsilon)$ of coefficients of the corresponding convex combination, that is,

$$y_n(\varepsilon) = \sum_{i \in I_n} \alpha_n^{(i)}(\varepsilon) x_i, \quad \sum_{i \in I_n} \alpha_n^{(i)}(\varepsilon) = 1, \quad \alpha_n^{(i)}(\varepsilon) \ge 0 \quad \forall i \in I_n.$$

Due to finite precision of computations, even in the case when the vectors x_i, $i \in I_n$, are affinely independent, all coefficients $\alpha_n^{(i)}(\varepsilon)$, $i \in I_n$, might be nonzero. Therefore we propose to use the following heuristic rule for choosing a point $x_{i_{1n}}$, $i_{1n} \in I_n$, that is removed from the polytope P_n by the steepest descent exchange rule. Namely, we choose as i_{1n} any index from I_n on which the minimum in $\min_{i \in I_n} \alpha_n^{(i)}(\varepsilon)$ is attained and add a safeguard based on the index removal method, discussed above, to ensure the validity of a certain approximate distance decay condition. Note, however, that for some methods (such as the Wolfe method [32]) there always exists $i \in I_n$ such that $\alpha_n^{(i)}(\varepsilon) = 0$, even when the computations are performed with finite precision. For such methods the safeguard based on the index removal method is completely unnecessary.

Thus, we arrive at the following robust version of Meta-algorithm 1 given below in Meta-algorithm 2. This meta-algorithm checks the approximate distance decay condition

$$\|y_{n+1}(\varepsilon) - z\| < \|y_n(\varepsilon) - z\|$$

to verify the correctness of the index exchange. If the condition fails, one needs to rectify the choice of the index i_{1n} on the previous step (that is, a wrong point was removed from P_n and one must remove a different point).

To correct the choice of i_{1n}, the meta-algorithm first computes the projection of z onto the affine hull $\operatorname{aff}\{x_i \mid i \in I_n\}$ of the points x_i, $i \in I_n$, on Step 3 (see problem (2.11)). Let us note that problem (2.11) can be reduced to the problem of solving a system of linear equations (see [32]).

As will be shown below, if the points x_i, $i \in I_n$, are affinely independent or the projection of z onto their affine hull does not coincide with the projection of z onto their convex hull, Step 3 makes a necessary correction of the point $y_n(\varepsilon)$ (more precisely, the coefficients $\alpha_n(\varepsilon)$ of the corresponding convex combination) to ensure that the new choice of i_{1n} on Step 1 leads to the validity of the approximate distance decay condition. Otherwise, the meta-algorithm moves to Step 4 and employs essentially the same technique as in the index removal method to correct the coefficients $\alpha_n(\varepsilon)$ and find the required index i_{1n}.

Let us analyze Meta-algorithm 2. First, we show that if $\min_{i \in I_n} \alpha_n^{(i)}(\varepsilon) = 0$ for some $n \in \mathbb{N}$ on Step 1 of this meta-algorithm, then under some natural assumptions no corrections of the coefficients $\alpha_n(\varepsilon)$ are needed, the method does not execute Steps 3 and 4, and moves to the next (i.e. $(n+1)$th) iteration.

As in the previous section, for any $n \in \mathbb{N}$ denote by y_n the actual projection of z onto P_n, that is, an optimal solution of the problem $\min_{y \in P_n} \|y - z\|$

Lemma 2.14. *Suppose that*

$$\max\Big\{2(\operatorname{diam}(P) + \operatorname{dist}(z, P))\varepsilon, \operatorname{diam}(P)\sqrt{\max\{0, 2\varepsilon\theta_0 - \varepsilon^2\}}\Big\} < \eta \le \operatorname{diam}(P)^2 \qquad (2.13)$$

and the algorithm $\mathscr{A}_\varepsilon$ with $\varepsilon \ge 0$ satisfies the following ***approximate optimality condition****: for any point $w \in \mathbb{R}^d$ and any polytope $Q = \operatorname{co}\{u_1, \ldots, u_m\} \subset \mathbb{R}^d$ one has*

$$\Big\| \sum_{i=1}^m \alpha^{(i)} u_i - u_* \Big\| < \varepsilon, \quad \sum_{i=1}^m \alpha^{(i)} = 1, \quad \alpha^{(i)} \ge 0 \quad \forall i \in \{1, \ldots, m\},$$

where $\alpha = \mathscr{A}_\varepsilon(w, Q)$ and u_* is an optimal solution of the nearest point problem $\min_{u \in Q} \|u - w\|$. Let also for some $n \in \mathbb{N}$ one has $\theta_{k+1} < \theta_k$ for any $k \in \{0, 1, \ldots, n-1\}$ and

$$\min_{i \in I_n} \alpha_n^{(i)}(\varepsilon) = 0 \qquad (2.14)$$

on Step 1 of Meta-algorithm 2. Then for θ_{n+1} computed on Step 2 of Meta-algorithm 2 one has $\theta_{n+1} < \theta_n$.

Algorithm 2 Robust meta-algorithm for finding the nearest point in a polytope.

Input: a point $z \in \mathbb{R}^d$, a collection of points $\{x_1, \ldots, x_\ell\} \subset \mathbb{R}^d$, an algorithm $\mathscr{A}_\varepsilon$, $\varepsilon > 0$, for solving the nearest point problem, and $\eta > 0$.

Step 0: Put $n = 0$, choose an index set $I_n \subseteq I$ with $|I_n| = d+1$, and define $P_n = \operatorname{co}\{x_i \mid i \in I_n\}$. Compute $\alpha_n(\varepsilon) = \mathscr{A}_\varepsilon(z, P_n)$, $y_n(\varepsilon) = \sum_{i \in I_n} \alpha_n^{(i)}(\varepsilon) x_i$, and $\theta_n = \|y_n(\varepsilon) - z\|$.

Step 1: If $y_n(\varepsilon)$ satisfies the condition

$$\langle y_n(\varepsilon) - z, x_i - y_n(\varepsilon) \rangle \ge -\eta \quad \forall i \in I, \tag{2.10}$$

return $y_n(\varepsilon)$. Otherwise, find $i_{1n} \in I_n$ and $i_{2n} \in I$ such that

$$\alpha_n^{(i_{1n})}(\varepsilon) = \min_{i \in I_n} \alpha_n^{(i)}(\varepsilon), \quad \langle y_n(\varepsilon) - z, x_{i_{2n}} \rangle = \min_{i \in I} \langle y_n(\varepsilon) - z, x_i \rangle.$$

Put

$$I_{n+1} = \Big(I_n \setminus \{i_{1n}\} \Big) \cup \{i_{2n}\}, \quad P_{n+1} = \operatorname{co}\Big\{ x_i \Bigm| i \in I_{n+1} \Big\}.$$

Step 2: Compute $\alpha_{n+1}(\varepsilon) = \mathscr{A}_\varepsilon(z, P_{n+1})$, $y_{n+1}(\varepsilon) = \sum_{i \in I_{n+1}} \alpha_{n+1}^{(i)}(\varepsilon) x_i$, and $\theta_{n+1} = \|y_{n+1}(\varepsilon) - z\|$. If $\theta_{n+1} < \theta_n$, set $n = n+1$ and go to **Step 1**. Otherwise, go to **Step 3**.

Step 3: Find an approximate solution β_n of the problem

$$\min_\beta \Big\| \sum_{i \in I_n} \beta^{(i)} x_i - z \Big\|^2 \quad \text{subject to} \quad \sum_{i \in I_n} \beta^{(i)} = 1. \tag{2.11}$$

Compute $h_n = \sum_{i \in I_n} \beta_n^{(i)} x_i$ and $\beta_{\min} = \min_{i \in I_n} \beta_n^{(i)}$. If $\beta_{\min} < 0$, find

$$\lambda = \min \left\{ \frac{\alpha_n^{(i)}}{\alpha_n^{(i)} - \beta_n^{(i)}} \,\middle|\, i \in I_n \colon \beta_n^{(i)} < 0 \right\}, \tag{2.12}$$

define $\alpha_n(\varepsilon) = (1-\lambda)\alpha_n(\varepsilon) + \lambda \beta_n$, $y_n(\varepsilon) = (1-\lambda) y_n(\varepsilon) + \lambda h_n$, and $\theta_n = \|y_n(\varepsilon) - z\|$, and go to **Step 1**. If $\beta_{\min} = 0$, define $\alpha_n(\varepsilon) = \beta_n$, $y_n(\varepsilon) = h_n$, $\theta_n = \|h_n - z\|$, and go to **Step 1**. If $\beta_{\min} > 0$, go to **Step 4**.

Step 4: Choose any $j \in I_n$, find an approximate least-squares solution γ_n of the system

$$\sum_{i \in I_n \setminus \{j\}} \gamma^{(i)} (x_i - x_j) = 0, \quad \sum_{i \in I_n \setminus \{j\}} \gamma^{(i)} = 1,$$

and set $\gamma_n^{(j)} = -\sum_{i \in I_n \setminus \{j\}} \gamma_n^{(i)}$. Compute

$$\lambda = \min \left\{ -\frac{\alpha_n^{(i)}}{\gamma_n^{(i)}} \,\middle|\, i \in I_n \colon \gamma_n^{(i)} < 0 \right\}.$$

Define $\alpha_n(\varepsilon) = \alpha_n(\varepsilon) + \lambda \gamma_n$, $y_n(\varepsilon) = \sum_{i \in I_n} \alpha_n(\varepsilon) x_i$, $\theta_n = \|y_n(\varepsilon) - z\|$, and go to **Step 1**.

Proof. Let us divide the proof into two parts. First, we show that $\theta_k \ge \varepsilon$ for any $k \in \{0, \ldots, n\}$ and then use this result to prove the statement of the lemma.

Part 1. Suppose by contradiction that $\theta_k < \varepsilon$ for some $k \in \{0, 1, \ldots, n\}$, i.e. $\|y_k(\varepsilon) - z\| < \varepsilon$. Let x_* be an optimal solution of the problem $(\mathscr{P})$. Then by definition $\|x_* - z\| \le \|y_k(\varepsilon) - z\| < \varepsilon$, which yields $\|x_* - y_k(\varepsilon)\| < 2\varepsilon$. Hence by Proposition 2.13 one has

$$\langle y_k(\varepsilon) - z, x_i - y_k(\varepsilon) \rangle \ge -2(\operatorname{diam}(P) + \operatorname{dist}(z, P))\varepsilon$$

Therefore by the first inequality in (2.13) the point $y_k(\varepsilon)$ satisfies the stopping criterion (2.10), which contradicts our assumption that the meta-algorithm computes i_{1n} on Step 1 of the nth iteration for $n \geq k$.

Part 2. By our assumption $\alpha_n^{(i_{1n})}(\varepsilon) = \min_{i \in I_n} \alpha_n^{(i)}(\varepsilon) = 0$. Therefore

$$y_n(\varepsilon) = \sum_{i \in I_n} \alpha_n^{(i)}(\varepsilon) x_i \in \operatorname{co}\Big\{ x_i \Bigm| i \in I_n \setminus \{i_{1n}\} \Big\},$$

which yields $y_n(\varepsilon) \in P_{n+1}$, where the set P_{n+1} is defined on Step 1.

By the definition of i_{2n} (see Step 1 of the meta-algorithm) one has

$$\langle y_n(\varepsilon) - z, x_{i_{2n}} - y_n(\varepsilon) \rangle = \min_{i \in I} \langle y_n(\varepsilon) - z, x_i - y_n(\varepsilon) \rangle \leq -\eta. \tag{2.15}$$

Define $x_n(t) = (1-t) y_n(\varepsilon) + t x_{i_{2n}}$. Clearly, $x_n(t) \in P_{n+1}$ for all $t \in [0,1]$, since, as was noted above, $y_n(\varepsilon) \in P_{n+1}$ and $x_{i_{2n}} \in P_{n+1}$ by the definition of P_{n+1} (see Step 2 of the meta-algorithm).

For any $t \in \mathbb{R}$ one has

$$\begin{aligned} f(t) := \|x_n(t) - z\|^2 &= \Big\| (y_n(\varepsilon) - z) + t(x_{i_{2n}} - y_n(\varepsilon)) \Big\|^2 \\ &= \|y_n(\varepsilon) - z\|^2 + 2t \langle y_n(\varepsilon) - z, x_{i_{2n}} - y_n(\varepsilon) \rangle + t^2 \|x_{i_{2n}} - y_n(\varepsilon)\|^2. \end{aligned}$$

Hence with the use of (2.15) one obtains

$$f(t) \leq \theta_n^2 - 2\eta t + \operatorname{diam}(P)^2 t^2 \quad \forall \alpha \geq 0.$$

The minimum in t of the right-hand side of this inequality is attained at $t_* = \eta / \operatorname{diam}(P)^2$. From the second inequality in (2.13) it follows that $t_* \in (0,1]$. Therefore, the point $x_n(t_*)$ belongs to P_{n+1} and

$$\min_{t \in [0,1]} f(t) = f(t_*) = \|x_n(t_*) - z\|^2 \leq \theta_n^2 - \frac{\eta^2}{\operatorname{diam}(P)^2}.$$

Applying the definition of y_{n+1}, the first inequality in (2.13), and the fact that $\theta_0 \geq \varepsilon$ one gets

$$\|y_{n+1} - z\|^2 = \min_{y \in P_{n+1}} \|y - z\| \leq \|x_n(t_*) - z\|^2 < \theta_n^2 - 2\theta_0 \varepsilon + \varepsilon^2.$$

Hence $\|y_{n+1} - z\|^2 < (\theta_n - \varepsilon)^2$, thanks to the fact that $\varepsilon \leq \theta_n < \theta_{n-1} < \ldots < \theta_0$ by our assumption and the first part of the proof. Consequently, by the approximate optimality condition on $\mathscr{A}_\varepsilon$, for $y_{n+1}(\varepsilon)$ computed on Step 2 one has

$$\theta_{n+1} = \|y_{n+1}(\varepsilon) - z\| \leq \|y_{n+1} - z\| + \varepsilon < \theta_n - \varepsilon + \varepsilon = \theta_n.$$

Thus, the approximate distance decay holds true, the meta-algorithm increments n on Step 2 and does not execute Steps 3 and 4. □

Remark 2.15. Let us underline that the lemma above holds true regardless of whether θ_0, θ_1, $\ldots$, θ_n, and $\alpha_n(\varepsilon)$ were computed on Step 2, 3 or 4. In particular, it holds true even if the equality (2.14) is satisfied for $\alpha_n(\varepsilon)$ that was computed on Step 3 or 4 and not directly computed by the algorithm $\mathscr{A}_\varepsilon$.

Remark 2.16. Note that the value θ_0 that is a priori unknown is used in inequalities (2.13) on parameters of Meta-algorithm 2. However, we can easily estimate it from above. If θ_0 is computed on Step 0, then by the approximate optimality condition $\theta_0 \leq \operatorname{dist}(z, P_0) + \varepsilon$. In turn, if θ_0 is computed on Steps 3 or 4, then under some natural assumptions one can show that $\theta_0 \leq \operatorname{dist}(z, P_0) + 2\varepsilon$ (see the proof of Theorem 2.20 below).

The previous lemma allows one to immediately prove correctness and finite termination of Meta-algorithm 2 in the case when the algorithm $\mathscr{A}_\varepsilon$ always returns a vector $\alpha_n(\varepsilon)$ having at least one zero component, regardless of whether the points x_i, $i \in I_n$, are affinely independent or not. Recall that this assumption is satisfied for the Wolfe method [32].

Theorem 2.17. *Let $\ell \ge d+1$, inequalities* (2.13) *hold true, and the algorithm $\mathscr{A}_\varepsilon$ with $\varepsilon > 0$ satisfy the approximate optimality condition from Lemma 2.14. Suppose also that for any point $w \in \mathbb{R}^d$ and any polytope $Q = \operatorname{co}\{u_1, \ldots, u_m\} \subset \mathbb{R}^d$ with $m \ge d+1$ there exists $i \in \{1, \ldots, m\}$ such that for $\alpha = \mathscr{A}_\varepsilon(w, Q)$ one has $\alpha^{(i)} = 0$. Then Meta-algorithm 2 is correctly defined, never executes Steps 3 and 4, terminates after a finite number of iterations, and returns a point $y_n(\varepsilon)$ such that $\|y_n(\varepsilon) - x_*\| < \sqrt{\eta}$, where x_* is an optimal solution of the problem $(\mathscr{P})$.*

Proof. From the assumptions of the theorem and Lemma 2.14 it follows that $\theta_1 < \theta_0$ and the meta-algorithm does not execute Steps 3 and 4 for $n = 0$, provided the stopping criterion (2.10) is not satisfied for $y_0(\varepsilon)$ (otherwise, the method terminates when $n = 0$ and does not execute Step 2).

Now, arguing by induction and applying Lemma 2.14 one can check that $\theta_n < \theta_{n-1}$ and the meta-algorithm does not execute Steps 3 and 4 for any $n \in \mathbb{N}$, if the stopping criterion (2.10) is not satisfied for $y_k(\varepsilon)$ with $k \in \{0, 1, \ldots, n-1\}$ (otherwise, the meta-algorithm never reaches nth iteration).

Thus, the meta-algorithm is correctly defined and the corresponding (finite or infinite) sequence $\{y_n(\varepsilon)\}$ satisfies the approximate distance decay condition:

$$\|y_n(\varepsilon) - z\| = \theta_n < \theta_{n-1} = \|y_{n-1}(\varepsilon) - z\|.$$

From this inequality it follows that all polytopes $P_0, P_1, \ldots, P_n, \ldots$ are distinct. Recall that each P_i is the convex hull of $d+1$ points from the set $\{x_1, \ldots, x_\ell\}$. Since there is only a finite number of distinct $d+1$-point subsets of the set $\{x_1, \ldots, x_\ell\}$, one must conclude that after a finite number of iterations the stopping criterion (2.10) must be satisfied, that is, the meta-algorithm terminates after a finite number of iterations. Moreover, any point $y_n(\varepsilon)$ satisfying the stopping criterion also satisfies the inequality $\|y_n(\varepsilon) - x_*\| < \sqrt{\eta}$ by Proposition 2.13. □

Remark 2.18. Let us comment on the assumption (2.13) on parameters of the algorithm $\mathscr{A}_\varepsilon$ and Meta-algorithm 2. Roughly speaking, inequality (2.13) means that to solve the problem $(\mathscr{P})$ with a pre-specified accuracy $\eta > 0$ with the use of Meta-algorithm 2 one must assume that the algorithm $\mathscr{A}_\varepsilon$ solves the corresponding reduced nearest point subproblems $\min_{x \in P_n} \|x - z\|$ with higher accuracy. Qualitatively, condition (2.13) can be rewritten as $\eta = O(\sqrt{\varepsilon})$ and viewed as a mathematical formulation of an intuitively obvious fact that Meta-algorithm 2 has lower accuracy than the algorithm $\mathscr{A}_\varepsilon$ that is used as a subroutine on each iteration. However, when considered quantitatively, inequalities (2.13) seem to be too conservative. They can be slighly relaxed, if one uses a different stopping criterion of the form

$$\langle y_n(\varepsilon) - z, x_i - y_n(\varepsilon) \rangle \ge -\eta \|x_i - y_n(\varepsilon)\| \quad \forall i \in I.$$

Then arguing in the same way as in the proof of Lemma 2.14 one can check that it is sufficient to suppose that $\sqrt{2\varepsilon\theta_0 + \varepsilon^2} < \eta \le 1$.

Remark 2.19. Note that one can replace the approximate optimality condition on the algorithm $\mathscr{A}_\varepsilon$ from Lemma 2.14 by the following condition: for any point $w \in \mathbb{R}^d$ and any polytope

$Q = \operatorname{co}\{u_1, \ldots, u_m\} \subset \mathbb{R}^d$ the point $y = \sum_{i=1}^m \alpha^{(i)} u_i$ with $\alpha = \mathscr{A}_\varepsilon(w, Q)$ satisfies the inequality

$$\langle y - z, u_i - y \rangle \geq -\varepsilon \quad \forall i \in \{1, \ldots, m\},$$

that is, the algorithm $\mathscr{A}_\varepsilon$ returns a point approximately satisfying the optimality conditions for the problem $\min_{u \in Q} \|u - w\|$. If u_* is an optimal solution of this problem, then by Proposition 2.13 one has $\|\mathscr{A}_\varepsilon(w, Q) - u_*\| \leq \sqrt{\varepsilon}$. Consequently, the theorem above remains to hold true in this case, provided ε is replaced by $\sqrt{\varepsilon}$ in the first inequality in (2.13).

Let us now prove correctness and finite termination of Meta-algorithm 2 in the general case.

Theorem 2.20. *Let $\ell \geq d + 1$, inequalities* (2.13) *be satisfied, and the following approximate optimality conditions hold true:*

(1) for any point $w \in \mathbb{R}^d$ and any polytope $Q = \operatorname{co}\{u_1, \ldots, u_m\} \subset \mathbb{R}^d$ one has

$$\Big\| \sum_{i=1}^m \alpha^{(i)} u_i - u_* \Big\| < \varepsilon, \quad \sum_{i=1}^m \alpha^{(i)} = 1, \quad \alpha^{(i)} \geq 0 \quad \forall i \in \{1, \ldots, m\},$$

where $\alpha = \mathscr{A}_\varepsilon(w, Q)$ and u_* is an optimal solution of the problem $\min_{u \in Q} \|u - w\|$;

(2) if for some $n \in \mathbb{N}$ the meta-algorithm executes Step 3, then $\sum_{i \in I_n} \beta_n^{(i)} = 1$ and the inequality $\|h_n - z\| \leq \|y_n(\varepsilon) - z\|$ holds true;

(3) if for some $n \in \mathbb{N}$ the vectors x_i, $i \in I_n$, are affinely independent and the meta-algorithm executes Step 3, then $\beta_{\min} \leq 0$;

(4) if for some $n \in \mathbb{N}$ the meta-algorithm executes Step 4, then

$$\Big\| \sum_{i \in I_n \setminus \{j\}} \gamma^{(i)} (x_i - x_j) \Big\| < \frac{\theta_{n-1} - \theta_n}{\lambda}, \quad \sum_{i \in I_n \setminus \{j\}} \gamma^{(i)} \neq 0$$

where $\lambda > 0$ is computed on Step 4 (if $n = 0$, then only the second inequality should be satisfied).

Then Meta-algorithm 2 is correctly defined, executes Steps 3 and 4 at most once per iteration, terminates after a finite number of iterations, and returns a point $y_n(\varepsilon)$ satisfying the inequality $\|y_n(\varepsilon) - x_*\| < \sqrt{\eta}$, where x_* is an optimal solution of the problem $(\mathscr{P})$.

Proof. Firstly, let us note that if the following two conditions hold true:

(1) the meta-algorithm is correctly defined (that is, there are no infinite loops involving Steps 3 and 4),

(2) the updating of θ_n on Steps 3 and 4 preserves the condition $\theta_n < \theta_{n-1}$,

then the meta-algorithm generates a finite or infinite sequence $\{y_n(\varepsilon)\}$ satisfying the approximate distance decay condition:

$$\theta_0 > \theta_1 > \ldots > \theta_n > \ldots \tag{2.16}$$

(recall that $\theta_n = \|y_n(\varepsilon) - z\|$). Indeed, according to the description of the method (see Meta-algorithm 2), the meta-algorithm increments n and moves to the next iteration if and only if the condition $\theta_{n+1} < \theta_n$ is satisfied on Step 2. Otherwise, it moves to Steps 3 and 4, corrects $\alpha_n(\varepsilon)$ and $y_n(\varepsilon)$, and repeats Steps 1 and 2 till the condition $\theta_{n+1} < \theta_n$ is satisfied. Thus, if (i) there are no infinite loops involving Steps 3 and 4, and (ii) the condition $\theta_n < \theta_{n-1}$ is preserved after updating θ_n on Steps 3 and 4, then inequalities (2.16) hold true.

Secondly, as was noted in the proof of Theorem 2.17, the validity of the approximate distance decay condition (2.16) implies that all polytopes $P_0, P_1, \ldots, P_n, \ldots$ are distinct. Hence taking into account the facts that each P_i is the convex hull of $d+1$ points from the set $\{x_1, \ldots, x_\ell\}$ and there is only a finite number of distinct $d+1$-point subsets of $\{x_1, \ldots, x_\ell\}$, one must conclude that after a finite number of iterations the stopping criterion (2.10) must be satisfied. Moreover, any point $y_n(\varepsilon)$ satisfying this criterion also satisfies the inequality $\|y_n(\varepsilon) - x_*\| < \sqrt{\eta}$ by Proposition 2.13.

Thus, to complete the proof of the theorem we need to prove that (i) there are no infinite loops involving Steps 3 and 4, and (ii) the updating of θ_n on Steps 3 and 4 does not break the condition $\theta_n < \theta_{n-1}$. In addition, our aim is to prove a slightly stronger statement that Steps 3 and 4 are executed at most once per iteration. Let us prove all these statements by induction.

Since the proof of the case $n = 0$ is essentially the same as the proof of the inductive step, we will consider only the inductive step. Fix any $n \in \mathbb{N}^*$ and suppose that for any $k \in \{0, 1, \ldots, n-1\}$ Steps 3 and 4 were performed at most once during the kth iteration and the condition

$$\theta_0 > \theta_1 > \ldots > \theta_{n-2} > \theta_{n-1} > \theta_n \tag{2.17}$$

holds true.

Clearly, we only need to consider the case when the meta-algorithm executes Step 3 on iteration n and Step 3 has not been executed before on this iteration. In this case, according to the scheme of the meta-algorithm $y_n(\varepsilon) = \sum_{i \in I_n} \alpha_n^{(i)}(\varepsilon) x_i$ with $\alpha_n(\varepsilon) = \mathscr{A}_\varepsilon(z, P_n)$ and the point $y_n(\varepsilon)$ does not satisfy the stopping criterion (2.10) (see Steps 1–4). Let us check that $z \notin P_n$.

Indeed, suppose that $z \in P_n$. Then $y_n = z$ is an optimal solution of the problem $\min_{y \in P} \|y - z\|$. Observe that by the first approximate optimality condition one has $\|y_n - y_n(\varepsilon)\| \le \varepsilon$. Therefore by Proposition 2.13 one has

$$\langle y_n(\varepsilon) - z, x_i - y_n(\varepsilon) \rangle \ge -\operatorname{diam}(P_n)\varepsilon \ge -\operatorname{diam}(P)\varepsilon \quad \forall i \in I.$$

Hence taking into account the first inequality in (2.13) one can conclude that $y_n(\varepsilon)$ satisfies the stopping criterion (2.10), which is impossible.

Thus, $z \notin P_n$. Note also that $\min_{i \in I_n} \alpha_n(\varepsilon) > 0$, since otherwise by Lemma 2.14 the meta-algorithm does not execute Step 3.

Recall that $h_n = \sum_{i \in I_n} \beta_n^{(i)} x_i$ is the approximate projection of z onto $\operatorname{aff}\{x_i \mid i \in I_n\}$ computed on Step 3 and $\beta_{\min} = \min_{i \in I_n} \beta_n^{(i)}$. Let us consider three cases.

Case I. Let $\beta_{\min} < 0$. Observe that for λ defined in (2.12) one has $\lambda \in (0,1)$, since $\alpha_n^{(i)}(\varepsilon) > 0$ for all $i \in I_n$. Define $\xi_n = (1-\lambda)\alpha_n(\varepsilon) + \lambda \beta_n$ and $w_n = (1-\lambda) y_n(\varepsilon) + \lambda h_n$. Then

$$\sum_{i \in I_n} \xi_n^{(i)} = (1-\lambda) \sum_{i \in I_n} \alpha_n^{(i)}(\varepsilon) + \lambda \sum_{i \in I_n} \beta_n^{(i)} = (1-\lambda) + \lambda = 1$$

(here the penultimate equality holds true by the first and second approximate optimality conditions). Moreover, if $\beta_n^{(i)} \ge 0$, then obviously $\xi_n^{(i)} \ge 0$, while if $\beta_n^{(i)} < 0$, then

$$\xi_n^{(i)} = (\alpha_n^{(i)}(\varepsilon) - \beta_n^{(i)}) \left(\frac{\alpha_n^{(i)}(\varepsilon)}{\alpha_n^{(i)}(\varepsilon) - \beta_n^{(i)}} - \lambda \right) \ge 0$$

by the definition of λ. In addition, for any i on which the minimum in the definition of λ is attained (see (2.12)) one has $\xi_n^{(i)} = 0$. Hence the point $w_n = (1-\lambda) y_n(\varepsilon) + \lambda h_n$ is a convex combination of the vectors x_i, $i \in I_n$, that is, $w_n \in P_n$. Furthermore, by the second approximate optimality condition one has

$$\|w_n - z\| \le (1-\lambda)\|y_n(\varepsilon) - z\| + \lambda\|h_n - z\| \le \|y_n(\varepsilon) - z\| = \theta_n.$$

Thus, after updating $\alpha_n(\varepsilon)$, $y_n(\varepsilon)$, and θ_n on Step 3 one has

$$\sum_{i \in I_n} \alpha_n^{(i)}(\varepsilon) = 1, \quad \min_{i \in I_n} \alpha_n^{(i)}(\varepsilon) = 0, \quad \theta_n < \theta_{n-1}. \tag{2.18}$$

Consequently, by Lemma 2.14 after performing Step 1 and computing $\alpha_{n+1}(\varepsilon) = \mathscr{A}_\varepsilon(z, P_{n+1})$ on Step 2 one has $\theta_{n+1} < \theta_n$. Thus, the meta-algorithm increments n and moves to the next iteration. In other words, in the case $\beta_{\min} < 0$ Step 3 of the meta-algorithm is performed only once and the condition $\theta_n < \theta_{n-1}$ is preserved.

Case II. Let $\beta_{\min} = 0$. Then $h_n \in P_n$. Moreover, by the second optimality condition one has $\|h_n - z\| \le \|y_n(\varepsilon) - z\|$, which implies that after setting $\theta_n = \|h_n - z\|$ the condition $\theta_n < \theta_{n-1}$ is preserved. In addition, for the updated value of $\alpha_n(\varepsilon) = \beta_n(\varepsilon)$ one has $\min_{i \in I_n} \alpha_n^{(i)}(\varepsilon) = 0$. Therefore, by Lemma 2.14 one can conclude that on Step 2 the condition $\theta_{n+1} < \theta_n$ is satisfied. Thus, in the case $\beta_{\min} = 0$ the meta-algorithm executes Step 3 only once and then after performing Steps 1 and 2 moves to the next iteration.

Case III. Let $\beta_{\min} > 0$. In this case the points x_i, $i \in I_n$, are affinely dependent by the third approximate optimality condition, and the meta-algorithm moves to Step 4. Let γ_n and λ be computed on Step 4. Recall that by definitions

$$\lambda = \min\left\{ -\frac{\alpha_n^{(i)}}{\gamma_n^{(i)}} \;\middle|\; i \in I_n \colon \gamma_n^{(i)} < 0 \right\} > 0.$$

and $\sum_{i \in I_n} \gamma_n^{(i)} = 0$.

Define $\xi_n = \alpha_n(\varepsilon) + \lambda\gamma_n$. Then $\sum_{i \in I_n} \xi_n^{(i)} = 1$, for any $i \in I_n$ such that $\gamma_n^{(i)} \ge 0$ one has $\xi_n^{(i)} \ge 0$, while for any $i \in I_n$ such that $\gamma_n^{(i)} < 0$ one has

$$\xi_n^{(i)} = -\gamma_n^{(i)}\left(-\frac{\alpha_n^{(i)}(\varepsilon)}{\gamma_n^{(i)}} - \lambda \right) \ge 0.$$

In addition, for any $i \in I_n$ on which the minimum in the definition of λ is attained one has $\xi_n^{(i)} = 0$. Note finally that by the fourth approximate optimality condition

$$\begin{aligned} \Big\| \sum_{i \in I_n} \xi_n^{(i)} x_i - z \Big\| &= \Big\| \sum_{i \in I_n} \alpha_n^{(i)}(\varepsilon) x_i - z + \lambda \sum_{i \ne j} \gamma_n^{(i)}(x_i - x_j) \Big\| \\ &\le \|y_n(\varepsilon) - z\| + \lambda \Big\| \sum_{i \ne j} \gamma_n^{(i)}(x_i - x_j) \Big\| < \theta_n + \lambda \frac{\theta_{n-1} - \theta_n}{\lambda} = \theta_{n-1}. \end{aligned}$$

Therefore, after an update on Step 4, values of $\alpha_n(\varepsilon)$, $y_n(\varepsilon)$, and θ_n satisfy conditions (2.18), which thanks to Lemma 2.14 implies that after performing Steps 1 and 2 the meta-algorithm increments n and moves to the next iteration. In other words, in the case $\beta_{\min} > 0$ the meta-algorithm executes Steps 3 and 4 only once and then moves to the next iteration. Furthermore, the condition $\theta_n < \theta_{n-1}$ is preserved in this case as well. $\square$

Remark 2.21. Let us comment on the approximate optimality conditions from the previous theorem:
(i) The first condition simply states that the algorithm $\mathscr{A}_\varepsilon(w,Q)$ always returns a point from Q that lies in the ε neighbourhood of the projection of w onto Q. In turn, the fourth condition indicates the accuracy with which an approximate least squares solution on Step 4 should be computed. Note that while the second equality in

$$\sum_{i \in I_n \setminus \{j\}} \gamma^{(i)}(x_i - x_j) = 0, \qquad \sum_{i \in I_n \setminus \{j\}} \gamma^{(i)} = 1$$

is essentially irrelevant, as long as the sum of $\gamma^{(i)}$, $i \in I_n \setminus \{j\}$, is nonzero, the first equality must be solved with high enough accuracy to ensure that after updating $y_n(\varepsilon)$ the inequality $\theta_n < \theta_{n-1}$ still holds true.
(ii) The second approximate optimality condition states that the approximate distance to the affine hull x_i, $i \in I_n$, computed by a subroutine on Step 3 of Meta-algorithm 2, does not exceed the approximate distance to the convex hull of these points computed by the algorithm $\mathscr{A}_\varepsilon$. Roughly speaking, the second approximate optimality condition means that an approximate projection of z onto the affine hull $\operatorname{aff}\{x_i \mid i \in I_n\}$, is computed on Step 3 with at least the same accuracy as the algorithm $\mathscr{A}_\varepsilon$ computes an approximate projection of z onto the convex hull $P_n = \operatorname{co}\{x_n \mid i \in I_n\}$.
(iii) The third approximate optimality condition is needed to exclude some degenerate cases. It should be noted that in the ideal case when $\varepsilon = 0$, this assumption is not needed. Indeed, if the points x_i, $i \in I_n$, are affinely independent, then their affine hull coincides with the entire space $\mathbb{R}^d$. Hence taking into account the fact that $z \notin P_n$ (otherwise, the stopping criterion (2.10) would have been satisfied) one can conclude that $\beta_{\min} < 0$. However, when computations are performed with finite precision, for some highly degenerate problems one might have $\beta_{\min} > 0$, even in the case when x_i, $i \in I_n$, are affinely independent, due to computational errors. In such cases the method might get stuck in an infinite loop of correcting the coefficients $\alpha_n(\varepsilon)$. The third approximate optimality condition excludes such situations. It should be noted that a foolproof version of Meta-algorithm 2 must keep track of whether a correction of $\alpha_n(\varepsilon)$ on Steps 3 and 4 has already been attempted and send an error message, if the method tries to correct the coefficients the second time.

2.4. **Numerical experiments.**

The proposed acceleration technique was verified numerically via multiple experiments with various values of d and ℓ. For each choice of d and ℓ we randomly generated 10 problems and average computation time for these problems was used as a performance measure.

The problem data was chosen as follows. First, we randomly generated ℓ points $\{\widehat{x}_1, \ldots, \widehat{x}_\ell\}$ in $\mathbb{R}^d$, uniformly distributed over the d-dimensional cube $[-1,1]^d$. Then these points were compressed and shifted as follows:

$$x_i = (1 + 0.01\widehat{x}_i^{(1)}, \widehat{x}_i^{(2)}, \ldots, \widehat{x}_i^{(d)}) \quad \forall i \in \{1, \ldots, \ell\}.$$

The point z was chosen as $z = 0$. As was noted in [32] and demonstrated by our numerical experiments, this particular problem is very challenging for methods for finding the nearest point in a polytope (especially in the cases when either ℓ is much greater than d or d is large) both in terms of computation time and finding an optimal solution with high accuracy. For our numerical experiments we used the values $d = 3$, $d = 10$, and $d = 50$, while values of ℓ were chosen from the set

$$\{100, 200, 300, 500, 1000, 2000, 3000, 5000, 10000, 20000, 30000, 50000\}$$

and depended on d.

Remark 2.22. Let us note that we performed numerical experiments for many other values of d and ℓ, as well as, for many other types of problem data. Since the results of corresponding numerical experiments were qualitatively the same as the ones presented below, we do not include them here for the sake of shortness.

Without trying to conduct exhaustive numerical experiments, we tested our acceleration technique on 3 classic methods: the MDM method [24], the Wolfe method [32], and a method based on quadratic programming. All methods were implemented in MATLAB. The last method was based on solving the problem

$$\min_{\alpha} \frac{1}{2} \Big\| \sum_{i=1}^{\ell} \alpha^{(i)} x_i \Big\|^2 \quad \text{subject to} \quad \sum_{i=1}^{\ell} \alpha^{(i)} = 1, \quad \alpha^{(i)} \geq 0, \quad i \in I$$

with the use of `quadprog`, the standard MATLAB routine for solving quadratic programming problems. We used this routine with default settings. The number of iterations of the MDM and Wolfe method was limited to 10^6. We used the inequalities

$$\delta(v_k) < \varepsilon, \quad \langle X, P_J \rangle > \langle X, X \rangle - \varepsilon$$

with $\varepsilon = 10^{-4}$ as termination criteria for the MDM method and the Wolfe method respectively (see the descriptions of these methods in [24, 32]). The value 10^{-4} was used, since occasionally both methods failed to terminate with smaller value of ε for large ℓ (this statement was especially true for the MDM method).

Finally, we implemented each method "on its own" and also incorporated each method within the robust acceleration technique, that is, Meta-algorithm 2. The initial guess for the meta-algorithm was chosen as

$$I_0 = \{1, \ldots, d+1\}, \quad P_0 = \operatorname{co}\{x_1, \ldots, x_{d+1}\}.$$

To demonstrate that the estimate of η in Lemma 2.14 and Theorems 2.17 and 2.20 (see (2.13)) is very conservative, we used the value $\eta = 10^{-4}$ for $d = 3$ and $d = 10$, and $\eta = 5 \cdot 10^{-4}$ for $d = 50$, since the acceleration technique occasionally failed to find a point satisfying the stopping criterion for $\eta = 10^{-4}$ in this case. In addition, we terminated the computations, if computation time exceeded 1 minute.

The results of numerical experiments are given in Figures 1–3. Let us first note that both the MDM method and its accelerated version were very slow and inaccurate in comparison with other methods in the case $d = 50$. That is why we do not present the corresponding results of the numerical experiments here. Furthermore, our numerical experiments showed that the difference in performance between the Wolfe method and its accelerated version significantly increases with the growth of d. Since the results were qualitatively the same for all d, here we present them only in the case $d = 50$, in which the difference in performance was the most noticeable.

The numerical experiments clearly demonstrate that the proposed acceleration technique with the steepest descent exchange rule allows one to significantly reduce the computation time for

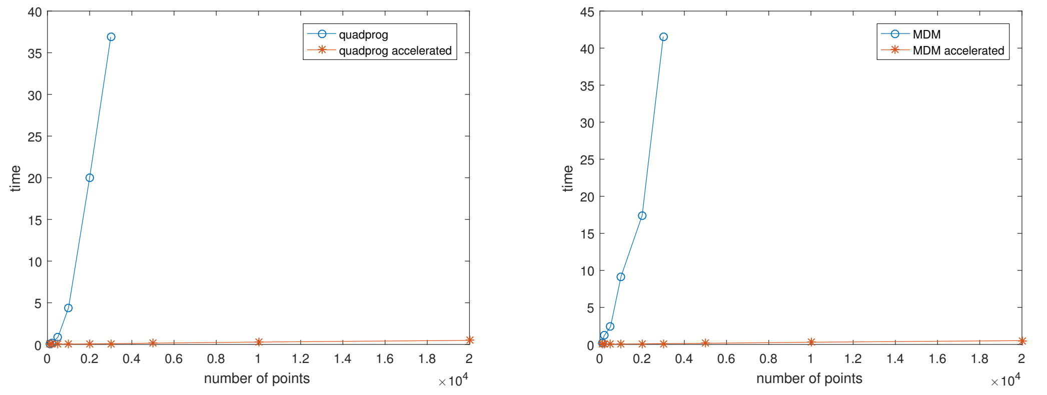


FIGURE 1. The results of numerical experiments in the case $d = 3$ for `quadprog` routine (left figure) and the MDM method (right figure).

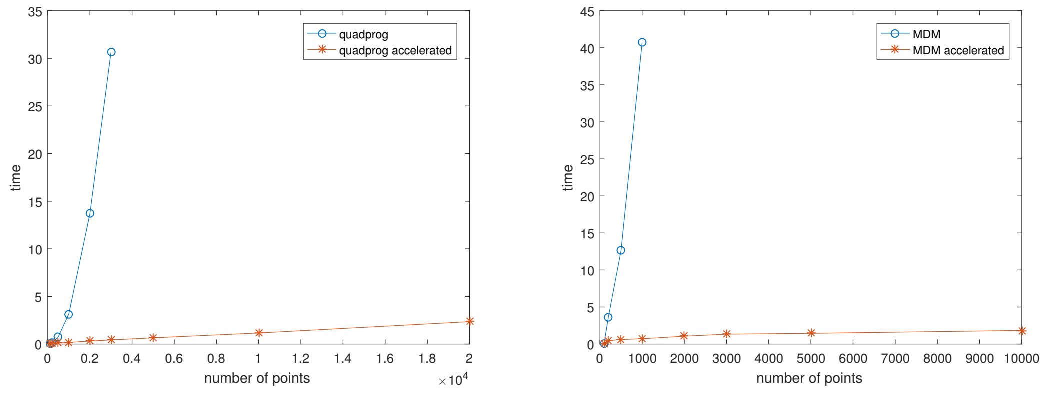


FIGURE 2. The results of numerical experiments in the case $d = 10$ for `quadprog` routine (left figure) and the MDM method (right figure).

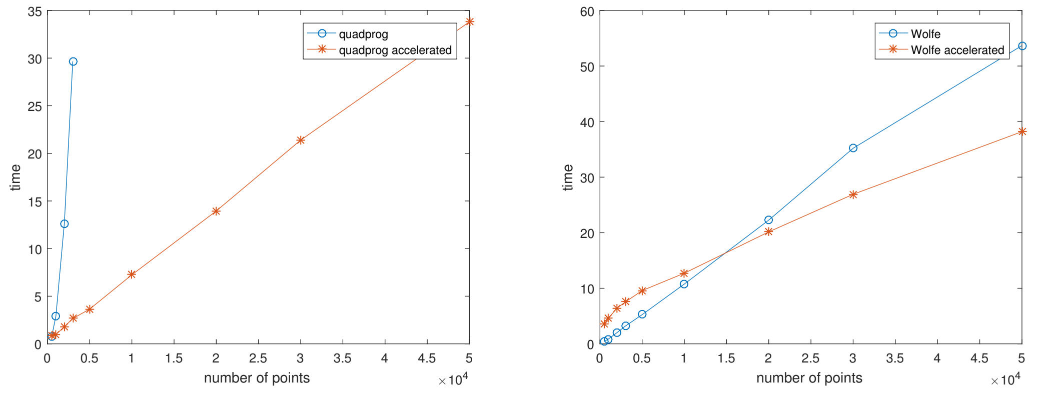


FIGURE 3. The results of numerical experiments in the case $d = 50$ for `quadprog` routine (left figure) and the Wolfe method (right figure).

methods of finding the nearest point in a polytope. In the case of `quadprog` routine and the

MDM method the reduction in time is proportional to the number of points ℓ. Moreover, numerical experiments also showed that for the accelerated versions of these methods the computation time increases *linearly* in ℓ for the problem under consideration (but we do not claim the linear in ℓ complexity of the acceleration technique for all types of problem data).

In the case of the Wolfe method the situation is somewhat different. For relatively small values of ℓ the "pure" Wolfe method outperforms its accelerated version, but for larger ℓ the accelerated version is faster than the pure method and the reduction of computation time increases with the increase of ℓ. However, it should be noted that precisely the same effect was observed for all other methods for finding the nearest point in a polytope and other types of problem data. When ℓ is close to d, it is faster to solve the corresponding problem with the use of the algorithm $\mathscr{A}_\varepsilon$, while when ℓ exceeds a certain threshold (depending on a particular method), the accelerated version of the algorithm $\mathscr{A}_\varepsilon$ starts to outperform the algorithm $\mathscr{A}_\varepsilon$ "on its own". Moreover, the reduction of the run-time increases with an increase of ℓ and is proportional to ℓ for large values of ℓ.

Thus, the main difference between the Wolfe method and other methods tested in our numerical experiments lies in the fact that the threshold value of ℓ, after which the acceleration technique becomes efficient, is significantly higher for the Wolfe method that for other methods.

Finally, let us note that the average number of iterations of Meta-algorithm 2, as expected, depended on the method to which this technique was applied. It was the highest for the MDM method, while the number of iterations of the meta-algorithm using the Wolfe method and `quadprog` routine was roughly the same. In the case $d = 3$ it was equal to 6, in the case $d = 10$ it was equal to 25.6, while in the case $d = 50$ it was equal to 150.8 (note that the problem is harder to solve for larger d). The results of other multiple numerical experiments, not reported here for the sake of shortness, showed that the average number of iterations of the meta-algorithm in most cases lies between d and $10d$ for various values of d and ℓ. Since the complexity of each iteration of the method is proportional to ℓ, the results of numerical experiments hint at the linear in ℓ average case complexity of Meta-algorithm 2, but we do not claim this complexity estimate to be true in the general case (especially, in the worst case) and are not ready to provide its theoretical justification.

It should be noted that an increase of average number of iterations with an increase of d is fully expected. To understand a reason behind it, observe that if the projection of z onto the polytope $P = \operatorname{co}\{x_1, \ldots, x_\ell\}$ belongs to the relative interior of a facet F of P, then to find this projection the meta-algorithm needs to find a subpolytope P_n containing at least d extreme point of F. If none of these points belongs to the initial guess $P_0 = \operatorname{co}\{x_1, \ldots, x_{d+1}\}$, then at at least d iterations are needed to find the required polytope P_n.

For example, if in the case $d = 2$ the polytope P_0 lies in the interior of P, then one needs at least 2 iterations for the polytope P_n to contain the edge of P to which the projection of z onto P belongs. In the case $d = 3$, the minimal number of iterations increases to 3, etc. Thus, the number of iterations of the meta-algorithm grows whenever d is increased.

Remark 2.23. In our implementation of Meta-algorithm 2, the algorithm $\mathscr{A}_\varepsilon$ was applied afresh on each iteration (i.e. without using any information from the previous iteration). It should be noted that, in particular, the performance of the accelerated version of the Wolfe method can be significantly improved, if one uses the *corral* (see [32]) computed on the previous iteration as the initial guess for the next iteration (a similar remark is true for accelerated versions of other methods). However, since our main goal was to demonstrate the performance of the acceleration technique on its own, here we do not discuss potential ways this technique can be efficiently integrated with a particular method for finding the nearest point in a polytope and do not present any results of numerical experiments for such fully integrated accelerated methods.

3. A comparison with the Wolfe method

Meta-algorithm 1 with the steepest descent exchange rule and Meta-algorithm 2 share many similarities with the Wolfe method [32] (and the Frank-Wolfe algorithms [20, 21]). Nonetheless, there is one important difference between these meta-algorithms and the Wolfe method, which, as the results of numerical experiments presented in the previous section demonstrate (see Figure 3), allows Meta-algorithm 2 to outperform the Wolfe method [32] in the case when the number of points is significantly greater than the dimension of the space.

The difference consists in the way in which the steepest descent exchange rule and the Wolfe method remove redundant points on each iteration. The Wolfe method operates with the so-called *corrals* (i.e. an affinely independent set of points from the polytope such that the projection of the origin onto the affine hull of this set belongs to the relative interior of its convex hull), while Meta-algorithm 1 with the steepest descent exchange rule and Meta-algorithm 2 operate with convex hulls of $d+1$ points without imposing any assumptions on them. On each iteration of the Wolfe method, given a current corral Q_n, one adds a new point x_n to this corral in the same way points are added in the steepest descent exchange rule, and then constructs a new corral Q_{n+1} from the set $\{x_n, Q_n\}$, filtering out multiple redundant points in the general case. In contrast, Meta-algorithm 1 with the steepest descent exchange rule and Meta-algorithm 2 remove only *one* point on each iteration.

This difference, apart from saving significant amount of time needed to find a corral in the spaces of moderate and large dimensions, also leads to a significantly different behaviour of Meta-algorithms 1 and 2 in comparison with the Wolfe method in the general case. This difference in behaviour occurs due to the fact that the projection of a given point onto the "unfiltered" subpolytope used by the meta-algorithms might be significantly different from the projection of a point onto the corral constructed by the Wolfe method.

In order to demonstrate how the different strategies of removing redundant points affect the performance of both methods, we present the average number of outer loops (iterations/shifts of the subpolytope) for Meta-algorithm 2 and the Wolfe method in the case $d = 50$ (see Figure 4). Observe that the number of iterations of the meta-algorithm is significantly smaller than the number of iterations of the Wolfe method, which explains why for large ℓ the meta-algorithm outperformes the Wolfe method in terms of computation time.

For the given problem data, the average number of iterations of the meta-algorithm stabilizes around 150 iterations for large ℓ, while in the case of the Wolfe method it stabilizes around 250 iterations for large ℓ. Thus, the completely different strategy of removing redundant points allows the meta-algorithm to reduce the number of iterations by about 40% in comparison with the Wolfe method.

4. Computing the distance between two polytopes

The acceleration technique for methods for finding the nearest point in a polytope from the previous section admits a straightforward extension to the case of methods for computing the

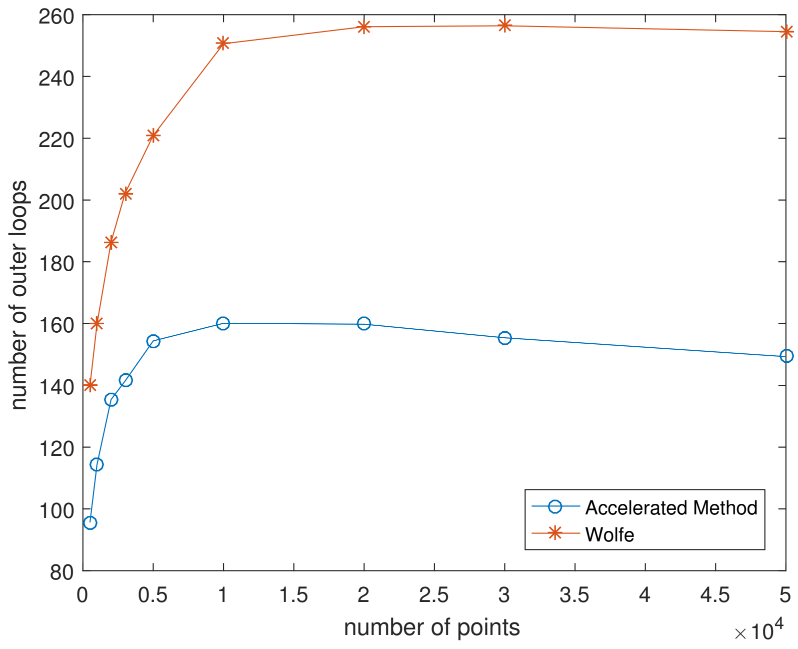


FIGURE 4. The average number of outer loops (iterations/shifts of the subpolytope) for Meta-algorithm 2 and the Wolfe method in the case $d = 50$.

distance between two convex polytopes defined as the convex hulls of finite sets of points. This section is devoted to a detailed discussion of such extension, that is, to a discussion of an acceleration technique for methods of solving the following optimization problem

$$\min_{(x,y)} \|x-y\| \quad \text{s.t.} \quad x \in P := \operatorname{co}\{x_1,\ldots,x_\ell\}, \ \ y \in Q = \operatorname{co}\{y_1,\ldots,y_m\}. \qquad (\mathscr{D})$$

in the case when $\ell \gg d$ and $m \gg d$. Here $x_i \in \mathbb{R}^d$, $i \in I = \{1,\ldots,\ell\}$, and $y_j \in \mathbb{R}^d$, $j \in J = \{1,\ldots,m\}$, are given points.

4.1. **An extension of the acceleration technique.**

Before we proceed to the description of an acceleration technique, let us present convenient optimality conditions for the problem $(\mathscr{D})$. These conditions are well-known, but we include their short proofs for the sake of completeness. Denote by $Pr_A(x)$ the Euclidean projection of a point $x \in \mathbb{R}^d$ on a closed convex set $A \subset \mathbb{R}^d$, i.e. an optimal solution of the problem $\min_{y\in A} \|x-y\|$.

Proposition 4.1. *A pair $(x_*, y_*) \in P \times Q$ is an optimal solution of the problem $(\mathscr{D})$ if and only if $Pr_P(y_*) = x_*$ and $Pr_Q(x_*) = y_*$ or, equivalently,*

$$\langle x_* - y_*, x_i - x_* \rangle \geq 0 \quad \forall i \in I, \quad \langle y_* - x_*, y_j - y_* \rangle \geq 0 \quad \forall j \in J. \qquad (4.1)$$

Proof. The fact that conditions (4.1) are equivalent to the validity of the equalities $Pr_P(y_*) = x_*$ and $Pr_Q(x_*) = y_*$ follows directly from Proposition 2.1. Let us check that these conditions are equivalent to the optimality of (x_*, y_*).

Indeed, as is easily seen, inequalities (4.1) are satisfied if and only if

$$\langle x_* - y_*, x - x_* \rangle \geq 0 \quad \forall x \in P, \quad \langle y_* - x_*, y - y_* \rangle \geq 0 \quad \forall y \in Q.$$

Algorithm 3 Meta-algorithm for finding the distance between two polytopes.

```
**Input:** two collection of points $\{x_1, \ldots, x_\ell\} \subset \mathbb{R}^d$ and $\{y_1, \ldots, y_m\} \subset \mathbb{R}^d$, an algorithm $\mathscr{A}$ for solving the problem of computing the distance between two polytopes, parameters $s, q \in \{1, \ldots, \min\{\ell, m\}\}$ with $s \ge q$, and an $(s,q)$-exchange rule $\mathscr{E}$.
**Initialization:** Put $n = 0$, choose index sets $I_n \subseteq I$ and $J_n \subseteq J$ with $|I_n| = |J_n| = s$. Define $P_n = \operatorname{co}\{x_i \mid i \in I_n\}$ and $Q_n = \operatorname{co}\{y_j \mid j \in J_n\}$.
**Step 1.** Compute $(v_n, w_n) = \mathscr{A}(P_n, Q_n)$ and
    $\rho_{xn} = \min_{i \in I} \langle v_n - w_n, x_i - v_n \rangle, \quad \rho_{yn} = \min_{j \in J} \langle w_n - v_n, y_j - w_n \rangle.$
If $\rho_{xn} \ge 0$ and $\rho_{yn} \ge 0$, **return** $(v_n, w_n)$.
**Step 2.** If $\rho_{xn} < 0$, compute $(I_{1n}, I_{2n}) = \mathscr{E}(I_n)$ and define
    $I_{n+1} = \Big( I_n \setminus I_{1n} \Big) \cup I_{2n}, \quad P_{n+1} = \operatorname{co}\big\{ x_i \bigm| i \in I_{n+1} \big\}.$
Otherwise, set $I_{n+1} = I_n$ and $P_{n+1} = P_n$.
If $\rho_{yn} < 0$, compute $(J_{1n}, J_{2n}) = \mathscr{E}(J_n)$ and define
    $J_{n+1} = \Big( J_n \setminus J_{1n} \Big) \cup J_{2n}, \quad Q_{n+1} = \operatorname{co}\big\{ y_j \bigm| j \in J_{n+1} \big\}.$
Otherwise, define $J_{n+1} = J_n$ and $Q_{n+1} = Q_n$. Put $n = n + 1$ and go to **Step 1**.
```

In turn, these conditions are satisfied if and only if

$$\langle x_* - y_*, x - x_* \rangle + \langle y_* - x_*, y - y_* \rangle \ge 0 \quad \forall x \in P, \, y \in Q.$$

It remains to note that by the standard optimality conditions for convex programming problems the inequality above is equivalent to the optimality of (x_*, y_*). □

Suppose that an algorithm $\mathscr{A}$ for solving the problem $(\mathscr{D})$ is given. For any two polytopes $V, W \subset \mathbb{R}^d$, defined as the convex hulls of finite collections of points, it returns an optimal solution $(v_*, w_*) = \mathscr{A}(V, W)$ of the problem

$$\min_{(v,w)} \|v - w\| \quad \text{subject to} \quad v \in V, \quad w \in W.$$

We propose to accelerate this algorithm in precisely the same way as an algorithm for solving the nearest point problem. Namely, one chooses "small" subpolytopes $P_0 \subset P$ and $Q_0 \subset Q$ and applies the algorithm $\mathscr{A}$ to find the distance between these subpolytopes. If an optimal solution of this problem coincides with an optimal solution of the problem $(\mathscr{D})$ (this fact is verified via the optimality conditions from Proposition 4.1), then the computations are terminated. Otherwise, one shifts these polytopes and repeats the same procedure till an optimal solution of the distance problem for subpolytopes P_n and Q_n coincides with an optimal solution of the problem $(\mathscr{D})$. A general structure of this acceleration technique (meta-algorithm) is essentially the same as the structure of Meta-algorithm 1 and is given in Meta-algorithm 3. For the sake of simplicity we suppose that the same exchange rule is used for each polytope, and the subpolytopes P_n and Q_n are convex hulls of the same number of points.

Arguing in essentially the same way as in the proofs of Lemmas 2.4 and 2.6 one can verify that the following results hold true. These results can be viewed as criteria for choosing an efficient exchange rule for Meta-algorithm 3.

For any sets $A, B \subset \mathbb{R}^d$ denote by

$$\operatorname{dist}(A,B) = \inf\big\{ \|x - y\| \bigm| x \in A,\ y \in B \big\}$$

the Euclidean distance between these sets.

Lemma 4.2. *Suppose that the exchange rule $\mathscr{E}$ satisfies the distance decay condition for the problem $(\mathscr{D})$: if for some $n \in \mathbb{N}$ the pair (v_n, w_n) does not satisfy the optimality conditions $\rho_{xn} \geq 0$ and $\rho_{yn} \geq 0$, then*

$$\operatorname{dist}(P_{n+1}, Q_{n+1}) < \operatorname{dist}(P_n, Q_n). \tag{4.2}$$

Then Meta-algorithm 3 terminates after a finite number of steps and returns an optimal solution of the problem $(\mathscr{D})$.

Lemma 4.3. *Let an (s,q)-exchange rule $\mathscr{E}$ with $s \geq q$ satisfy the distance decay condition for the problem $(\mathscr{D})$ for any polytopes $U, W \subset \mathbb{R}^d$. Then $s \geq d+1$.*

As in the case of the nearest point problem, one can utilise the steepest descent exchange rule to shift polytopes P_n and Q_n on each iteration of Meta-algorithm 3. In the case of the problem $(\mathscr{D})$ this exchange rule is defined as follows (for the sake of shortness we describe it only for the polytope P_n).

- **Input:** an index set $I_n \subset I$ with $|I_n| = d+1$, the set $\{x_1, \ldots, x_\ell\}$, and the pair $(v_n, w_n) = \mathscr{A}(P_n, Q_n)$.
- **Step 1:** Find $i_{1n} \in I_n$ such that $v_n \in \operatorname{co}\{x_i \mid I_n \setminus \{i_{1n}\}\}$.
- **Step 2:** Find $i_{2n} \in I$ such that

$$\langle v_n - w_n, x_{i_{2n}} \rangle = \min_{i \in I} \langle v_n - w_n, x_i \rangle.$$

Return $(\{i_{1n}\}, \{i_{2n}\})$.

The following theorem, whose proof is similar to the proof of Theorem 2.8, shows that the steepest descent exchange rule satisfies the distance decay condition for the problem $(\mathscr{D})$.

Theorem 4.4. *For any polytopes $P = \operatorname{co}\{x_1, \ldots, x_\ell\}$ and $Q = co\{y_1, \ldots, y_m\}$ with $\ell \geq d+1$ and $m \geq d+1$ the steepest descent exchange rule satisfies the distance decay condition for the problem $(\mathscr{D})$.*

Proof. Suppose that for some $n \in \mathbb{N}$ the pair (v_n, w_n) does not satisfy the stopping criterion $\rho_{xn} \geq 0$ and $\rho_{yn} \geq 0$ from Meta-algorithm 3.

Suppose that $\rho_{xn} < 0$. By definition $(v_n, w_n) \in P_n \times Q_n$ is an optimal solution of the problem

$$\min_{(x,y)} \|x - y\| \quad \text{subject to} \quad x \in P_n, \quad y \in Q_n.$$

Hence by Proposition 4.1 one has $Pr_{P_n}(w_n) = v_n$. Moreover, by our assumption one has

$$\min_{i \in I} \langle v_n - w_n, x_i - v_n \rangle < 0,$$

which means that v_n does not satisfy the optimality condition for the nearest point problem $\min_{x \in P_{n+1}} \|x - w_n\|$. Therefore, almost literally repeating the proof of Theorem 2.8 one gets that

$$\operatorname{dist}(w_n, P_{n+1}) < \operatorname{dist}(w_n, P_n) = \|w_n - v_n\| = \operatorname{dist}(Q_n, P_n).$$

Similarly, if $\rho_{yn} < 0$, then

$$\operatorname{dist}(v_n, Q_{n+1}) < \operatorname{dist}(v_n, Q_n) = \|v_n - w_n\| = \operatorname{dist}(P_n, Q_n).$$

In either case one has

$$\operatorname{dist}(P_{n+1}, Q_{n+1}) \le \min\Big\{ \operatorname{dist}(w_n, P_{n+1}), \operatorname{dist}(v_n, Q_{n+1}) \Big\} < \operatorname{dist}(P_n, Q_n),$$

which means that the distance decay condition for the problem $(\mathscr{D})$ holds true. □

Corollary 4.5. *Let $\ell, m \ge d+1$. Then Meta-algorithm 3 with $s = d+1$, $q = 1$, and the steepest descent exchange rule terminates after a finite number of iterations and returns an optimal solution of the problem $(\mathscr{D})$.*

As in the case of the nearest point problem, it is easier to implement the steepest descent exchange rule for the problem $(\mathscr{D})$, when the algorithm $\mathscr{A}$ returns not an optimal solution $(v_*, w_*) = \mathscr{A}(V, W)$ of the problem

$$\min \|v - w\| \quad \text{subject to} \quad V = \operatorname{co}\{v_1, \ldots, v_{\ell_1}\}, \quad W = \operatorname{co}\{w_1, \ldots, w_{\ell_2}\},$$

but coefficients of the corresponding convex combinations, that is, vectors $\alpha \in \mathbb{R}^{\ell_1}$ and $\beta \in \mathbb{R}^{\ell_2}$ such that

$$v_* = \sum_{i=1}^{\ell_1} \alpha^{(i)} v_i, \quad \sum_{i=1}^{\ell_1} \alpha^{(i)} = 1, \quad \alpha^{(i)} \ge 0 \quad \forall i \in \{1, \ldots, \ell_1\},$$
$$w_* = \sum_{j=1}^{\ell_2} \beta^{(j)} w_j, \quad \sum_{j=1}^{\ell_2} \beta^{(j)} = 1, \quad \beta^{(j)} \ge 0 \quad \forall j \in \{1, \ldots, \ell_2\}.$$

In this case, if on iteration n of Meta-algorithm 3 one computes a pair of coefficients of convex combinations $(\alpha_n, \beta_n) = \mathscr{A}(P_n, Q_n)$, then one can obviously choose as an index $i_{1n} \in I_n$ of vector $x_{i_{1n}}$ that is removed from P_n any index $k \in I_n$ such that $\alpha_n^{(k)} = 0$. Similarly, one can choose as an index j_{1n} of a vector $y_{j_{1n}}$ that is removed from Q_n any index $k \in J_n$ such that $\beta_n^{(k)} = 0$.

If the polytopes P_n and Q_n intersect, then Meta-algorithm 3 terminates on iteration n. If they do not intersect, then the points v_n and w_n obviously lie on the boundaries of P_n and Q_n respectively. Hence by [35, Lemma 2.8] in the case when the points x_i, $i \in I_n$, are affinely independent, there exists at least one $k \in I_n$ such that $\alpha_n^{(k)} = 0$. Similarly, in the case when the points y_j, $j \in J_n$, are affinely independent, there exists at least one $k \in J_n$ such that $\beta_n^{(k)} = 0$. In this case one can easily find the required indices $i_{1n} \in I_n$ and $j_{1n} \in J_n$.

If either x_i, $i \in I_n$, or y_j, $j \in J_n$, are affinely dependent, then one can utilise the following obvious extension of *the index removal method* from Section 2.2. For the sake of shortness we describe in only in the case when $\rho_{xn} < 0$ and $\rho_{yn} < 0$.

- **Input:** index sets $I_n \subset I$, $J_n \subset J$ with $|I_n| = |J_n| = d+1$, the sets $\{x_1, \ldots, x_\ell\}$ and $\{y_1, \ldots, y_m\}$, and $(\alpha_n, \beta_n) = \mathscr{A}(P_n, Q_n)$.
- **Step 1:** Compute $\alpha_{\min} = \min_{i \in I_n} \alpha_n^{(i)}$. If $\alpha_{\min} = 0$, find $i_{1n} \in I_n$ such that $\alpha_n^{(i_{1n})} = 0$. Otherwise, choose any $k \in I_n$ and compute a least-squares solution γ_n of the system

$$\sum_{i \in I_n \setminus \{k\}} \gamma^{(i)}(x_i - x_k) = 0, \qquad \sum_{i \in I_n \setminus \{k\}} \gamma^{(i)} = 1$$

  and set $\gamma_n^{(k)} = -1$. Find an index $i_{1n} \in I_n$ on which the minimum in

$$\min \left\{ -\frac{\alpha_n^{(i)}}{\gamma_n^{(i)}} \;\middle|\; i \in I_n \colon \gamma_n^{(i)} < 0 \right\}$$

is attained.

- **Step 2:** Compute $\beta_{\min} = \min_{j \in J_n} \beta_n^{(i)}$. If $\beta_{\min} = 0$, find $j_{1n} \in J_n$ such that $\beta_n^{(j_{1n})} = 0$. Otherwise, choose any $k \in J_n$ and compute a least-squares solution λ_n of the system
$$\sum_{j \in J_n \setminus \{k\}} \lambda^{(j)}(y_j - y_k) = 0, \qquad \sum_{j \in J_n \setminus \{k\}} \lambda^{(j)} = 1$$
and set $\lambda_n^{(k)} = -1$. Find an index $j_{1n} \in J_n$ on which the minimum in
$$\min \left\{ -\frac{\beta_n^{(j)}}{\lambda_n^{(j)}} \;\middle|\; j \in J_n \colon \lambda_n^{(j)} < 0 \right\}$$
is attained and **return** (i_{1n}, j_{1n}).

Arguing in precisely the same way as in the proof of Proposition 2.10 one can verify that the index removal method correctly finds the required indices $i_{1n} \in I_n$ and $j_{1n} \in J_n$.

Proposition 4.6. *Suppose that for some $n \in \mathbb{N}$ the stopping criterion $\rho_{xn} \geq 0$ and $\rho_{yn} \geq 0$ of Meta-algorithm 3 does not hold true, and let $(i_{1n}, j_{1n}) \in I_n \times J_n$ be the output of the index removal method. Then*
$$v_n \in \operatorname{co}\left\{ x_i \;\middle|\; i \in I_n \setminus \{i_{1n}\} \right\}, \quad w_n \in \operatorname{co}\left\{ y_j \;\middle|\; j \in J_n \setminus \{j_{1n}\} \right\}.$$

Remark 4.7. Let us note that in the case when the number of points in only one polytope is much greater than d (say $\ell \gg d$), while for the other polytope it is comparable to d or even smaller than the dimension of the space, one can propose a natural modification of the acceleration technique presented in this section. Namely, instead of shifting subpolytopes P_n and Q_n in both polytopes P and Q one needs to shift only polytope P_n inside P and define $Q_n \equiv Q$. An analysis of such modification of Meta-algorithm 3 is straightforward and is left to the interested reader.

4.2. **A robust version of the meta-algorithm.** Let us also present a robust version of Meta-algorithm 3 that takes into account finite precision of computations and is more suitable for practical implementation than the original method. To this end, as in Subsection 2.3, suppose that instead of the "ideal" algorithm $\mathscr{A}$ its "approximate" version $\mathscr{A}_\varepsilon$, $\varepsilon > 0$, is given. For any two polytopes $V, W \subset \mathbb{R}^d$ the algorithm $\mathscr{A}_\varepsilon$ return an approximate (in some sense) solution of the problem
$$\min_{(x,y)} \|x - y\| \quad \text{subject to} \quad x \in V, \, y \in W.$$
To ensure finite termination of the acceleration technique based on the "approximate" algorithm $\mathscr{A}_\varepsilon$ one obviously needs to replace the optimality conditions for the problem $(\mathscr{D})$ (see Proposition 4.1), which are used as a stopping criterion, with approximate optimality conditions of the form
$$\langle v_* - w_*, x_i - v_* \rangle \geq -\eta \quad \forall i \in I, \quad \langle w_* - v_*, y_j - w_* \rangle \geq -\eta \quad \forall j \in J$$
with some small $\eta > 0$. The following proposition shows how these approximate optimality conditions are related to approximate optimality of (v_*, w_*).

Proposition 4.8. *Let (v_*, w_*) be an optimal solution of the problem $(\mathscr{D})$ and a pair $(v, w) \in P \times Q$ satisfy the inequalities*
$$\langle v - w, x_i - v \rangle \geq -\eta \quad \forall i \in I, \quad \langle w - v, y_j - w \rangle \geq -\eta \quad \forall j \in J \tag{4.3}$$

for some $\eta > 0$. Then

$$\|v - w - (v_* - w_*)\| \le \sqrt{2\eta}, \quad \|v - w\| \le \operatorname{dist}(P, Q) + \sqrt{2\eta}.$$

Conversely, let $(v, w) \in P \times Q$ be such that $\|v - w - (v_* - w_*)\| \le \varepsilon$ for some $\varepsilon > 0$. Then the pair (v, w) satisfies inequalities (4.3) *for any $\eta > 0$ such that $\eta \ge (\operatorname{diam}(P) + \operatorname{diam}(Q) + \operatorname{dist}(P, Q))\varepsilon$.*

Proof. Let a pair $(v, w) \in P \times Q$ satisfy inequalities (4.3) for some $\eta > 0$ and (v_*, w_*) be an optimal solution of the problem $(\mathscr{D})$. Observe that

$$\|v - w - (v_* - w_*)\|^2 = \langle v - w, v - v_* \rangle + \langle w - v, w - w_* \rangle - \langle v_* - w_*, v - w - (v_* - w_*) \rangle.$$

By Proposition 4.1 one has

$$\langle v_* - w_*, x - y - (v_* - w_*) \rangle \ge 0 \quad \forall x \in P,\, y \in Q,$$

while from inequalities (4.3) it obviously follows that

$$\langle v - w, x - v \rangle \ge -\eta \quad \forall x \in P, \quad \langle w - v, y - w \rangle \ge -\eta \quad \forall y \in Q.$$

Therefore

$$\|v - w - (v_* - w_*)\|^2 \le 2\eta,$$

which yields

$$\Big| \|v - w\| - \|v_* - w_*\| \Big| \le \|v - w - (v_* - w_*)\| \le \sqrt{2\eta}$$

or, equivalently, $\|v - w\| \le \operatorname{dist}(P, Q) + \sqrt{2\eta}$, since $\|v_* - w_*\| = \operatorname{dist}(P, Q)$.

Suppose now that $\|v - w - (v_* - w_*)\| \le \varepsilon$ for some $\varepsilon > 0$ and (v_*, w_*) is an optimal solution of the problem $(\mathscr{D})$. Adding and subtracting $v_* - w_*$ twice one gets that for any $(x, y) \in P \times Q$ the following equality holds true:

$$\begin{aligned} \langle v - w, x - y - (v - w) \rangle = & \langle v - w - (v_* - w_*), x - y - (v - w) \rangle \\ & + \langle v_* - w_*, x - y - (v_* - w_*) \rangle + \langle v_* - w_*, v_* - w_* - (v - w) \rangle. \end{aligned}$$

Note that the second term on the right-hand side of this equality is nonnegative by Proposition 4.1, while the first and the last terms can be estimated as follows:

$$\begin{aligned} \big| \langle v - w - (v_* - w_*), x - y - (v - w) \rangle \big| &\le \|v - w - (v_* - w_*)\| \Big(\|x - v\| + \|y - w\| \Big) \\ &\le \varepsilon \big(\operatorname{diam}(P) + \operatorname{diam}(Q) \big), \end{aligned}$$

$$\big| \langle v_* - w_*, v_* - w_* - (v - w) \rangle \big| \le \|v_* - w_*\| \|v_* - w_* - (v - w)\| \le \operatorname{dist}(P, Q)\varepsilon.$$

Consequently, one has

$$\langle v - w, x - y - (v - w) \rangle \ge -\Big(\operatorname{diam}(P) + \operatorname{diam}(Q) + \operatorname{dist}(P, Q) \Big)\varepsilon,$$

and the proof is complete. □

To formulate an implementable robust version of Meta-algorithm 3, suppose that the output $\mathscr{A}_\varepsilon(P_n, W_n)$ of the algorithm $\mathscr{A}_\varepsilon$ is not an approximately optimal solution $(v_n(\varepsilon), w_n(\varepsilon))$ of the problem

$$\min_{(x,y)} \|x - y\| \quad \text{subject to} \quad x \in P_n, \quad y \in Q_n,$$

but rather coefficients of the corresponding convex combinations, that is, a pair $(\alpha_n(\varepsilon), \beta_n(\varepsilon)) \in \mathbb{R}^{d+1} \times \mathbb{R}^{d+1}$ such that

$$v_n(\varepsilon) = \sum_{i \in I_n} \alpha_n^{(i)}(\varepsilon) x_i, \quad \sum_{i \in I_n} \alpha_n^{(i)}(\varepsilon) = 1, \quad \alpha_n^{(i)}(\varepsilon) \geq 0 \quad \forall i \in I_n,$$
$$w_n(\varepsilon) = \sum_{j \in J_n} \beta_n^{(j)}(\varepsilon) y_j, \quad \sum_{j \in J_n} \beta_n^{(j)}(\varepsilon) = 1, \quad \beta_n^{(j)}(\varepsilon) \geq 0 \quad \forall j \in J_n.$$

Since the algorithm $\mathscr{A}_\varepsilon$ computes only an approximately optimal solution, even in the case when x_i, $i \in I_n$, are affinely independent, and y_j, $j \in J_n$, are affinely independent, all coefficients $\alpha_n^{(i)}(\varepsilon)$ and $\beta_n^{(j)}(\varepsilon)$ might be strictly positive. Therefore we propose to utilize essentially the same strategy for removing indices from the sets I_n and J_n as is used in Meta-algorithm 2. The main goal of this strategy is to maintain the validity of the approximate distance decay condition

$$\theta_{n+1} := \|v_{n+1}(\varepsilon) - w_{n+1}(\varepsilon)\| < \|v_n(\varepsilon) - w_n(\varepsilon)\| =: \theta_n \quad \forall n.$$

As we will show below, this inequality guarantees finite termination of the robust meta-algorithm for finding the distance between two polytopes given in Meta-algorithm 4.

The meta-algorithm uses a heuristic rule for choosing indices $i_{1n} \in I_n$ and $j_{1n} \in J_n$ that are removed from the subpolytopes P_n and Q_n on each iteration. This rule consists in finding the minimal coefficients

$$\alpha_n^{(i_{1n})}(\varepsilon) = \min_{i \in I_n} \alpha_n^{(i)}(\varepsilon), \quad \beta_n^{(j_{1n})}(\varepsilon) = \min_{j \in J_n} \beta_n^{(j)}(\varepsilon)$$

If such choice of indices $i_{1n} \in I_n$ and $j_{1n} \in J_n$ ensures the validity of the inequality $\theta_{n+1} < \theta_n$ (the approximate distance decay condition), then the meta-algorithm increments n and moves to the next iteration. Otherwise, it employs the **coefficients correction method** to update $\alpha_n(\varepsilon)$ and $\beta_n(\varepsilon)$ in such a way that would guarantee the validity of the approximate distance decay condition. The coefficients correction method is described below:

- **Step 1:** If $\rho_{xn} \geq -\eta$, go to **Step 3**. Otherwise, find an approximate solution γ_n of the problem
$$\min_\gamma \Big\| \sum_{i \in I_n} \gamma^{(i)} x_i - w_n(\varepsilon) \Big\|^2 \quad \text{subject to} \quad \sum_{i \in I_n} \gamma^{(i)} = 1.$$
Compute $h_n = \sum_{i \in I_n} \gamma_n^{(i)} x_i$ and $\gamma_{\min} = \min_{i \in I_n} \gamma_n^{(i)}$. If $\gamma_{\min} < 0$, compute
$$\mu = \min \left\{ \frac{\alpha_n^{(i)}(\varepsilon)}{\alpha_n^{(i)}(\varepsilon) - \gamma_n^{(i)}} \;\middle|\; i \in I_n \colon \gamma_n^{(i)} < 0 \right\},$$
define $\alpha_n(\varepsilon) = (1-\mu)\alpha_n(\varepsilon) + \mu \gamma_n$ and
$$v_n(\varepsilon) = (1-\mu) v_n(\varepsilon) + \mu h_n, \quad \theta_n = \|v_n(\varepsilon) - w_n(\varepsilon)\|.$$
If $\gamma_{\min} = 0$, define $\alpha_n(\varepsilon) = \gamma_n$, $v_n(\varepsilon) = h_n$, $\theta_n = \|h_n - w_n(\varepsilon)\|$, and go to **Step 3**. If $\gamma_{\min} > 0$, go to **Step 2**.
- **Step 2:** Choose any $k \in I_n$, find an approximate least-squares solution λ_n of the system
$$\sum_{i \in I_n \setminus \{k\}} \lambda^{(i)} (x_i - x_k) = 0, \quad \sum_{i \in I_n \setminus \{k\}} \lambda^{(i)} = 1$$

Algorithm 4 Robust meta-algorithm for finding the distance between two polytopes.

```
**Input:** two collection of points $\{x_1,\ldots,x_\ell\}, \{y_1,\ldots,y_m\} \subset \mathbb{R}^d$, $\eta > 0$, and an algorithm $\mathscr{A}_\varepsilon$, $\varepsilon > 0$, for finding the distance between two polytopes.
**Step 0:** Put $n = 0$, choose index sets $I_n \subseteq I$ and $J_n \subseteq J$ with $|J_n| = |I_n| = d+1$, and define
    $P_n = \operatorname{co}\{x_i \mid i \in I_n\}, \quad Q_n = \operatorname{co}\{y_j \mid j \in J_n\}.$
Compute $(\alpha_n(\varepsilon), \beta_n(\varepsilon)) = \mathscr{A}_\varepsilon(P_n, Q_n)$,
    $v_n(\varepsilon) = \sum_{i \in I_n} \alpha_n^{(i)}(\varepsilon) x_i, \quad w_n(\varepsilon) = \sum_{j \in J_n} \beta_n^{(j)}(\varepsilon) y_j, \quad \theta_n = \|v_n(\varepsilon) - w_n(\varepsilon)\|.$
**Step 1:** Compute
    $\rho_{xn} = \min_{i \in I} \langle v_n(\varepsilon) - w_n(\varepsilon), x_i - v_n(\varepsilon) \rangle, \quad \rho_{yn} = \min_{j \in J} \langle w_n(\varepsilon) - v_n(\varepsilon), y_j - w_n(\varepsilon) \rangle.$
If $\rho_{xn} \geq -\eta$ and $\rho_{yn} \geq -\eta$, **return** $(v_n(\varepsilon), w_n(\varepsilon))$. If $\rho_{xn} < -\eta$, find $i_{1n} \in I_n$ and $i_{2n} \in I$ such that
    $\alpha_n^{(i_{1n})}(\varepsilon) = \min_{i \in I_n} \alpha_n^{(i)}(\varepsilon), \quad \langle v_n(\varepsilon) - w_n(\varepsilon), x_{i_{2n}} \rangle = \min_{i \in I} \langle v_n(\varepsilon) - w_n(\varepsilon), x_i \rangle,$
and define
    $I_{n+1} = \Big( I_n \setminus \{i_{1n}\} \Big) \cup \{i_{2n}\}, \quad P_{n+1} = \operatorname{co}\Big\{ x_i \Bigm| i \in I_{n+1} \Big\}.$
If $\rho_{yn} < -\eta$, find $j_{1n} \in J_n$ and $j_{2n} \in J$ such that
    $\beta_n^{(j_{1n})}(\varepsilon) = \min_{j \in J_n} \beta_n^{(j)}(\varepsilon), \quad \langle w_n(\varepsilon) - v_n(\varepsilon), y_{j_{2n}} \rangle = \min_{j \in J} \langle w_n(\varepsilon) - v_n(\varepsilon), y_j \rangle,$
and define
    $J_{n+1} = \Big( J_n \setminus \{j_{1n}\} \Big) \cup \{j_{2n}\}, \quad Q_{n+1} = \operatorname{co}\Big\{ y_j \Bigm| j \in J_{n+1} \Big\}$
**Step 2:** Compute $(\alpha_{n+1}(\varepsilon), \beta_{n+1}(\varepsilon)) = \mathscr{A}_\varepsilon(P_{n+1}, Q_{n+1})$,
    $v_{n+1}(\varepsilon) = \sum_{i \in I_{n+1}} \alpha_{n+1}^{(i)}(\varepsilon) x_i, \quad w_{n+1}(\varepsilon) = \sum_{j \in J_{n+1}} \beta_{n+1}^{(j)}(\varepsilon) y_j,$
and $\theta_{n+1} = \|v_{n+1}(\varepsilon) - w_{n+1}(\varepsilon)\|$. If $\theta_{n+1} < \theta_n$, set $n = n+1$ and go to **Step 1**. Otherwise, go to **Step 3**.
**Step 3:** Apply the coefficients correction method and go to Step 1.
```

and set $\lambda_n^{(k)} = -\sum_{i \in I_n \setminus \{k\}} \lambda_n^{(i)}$. Compute

$$\nu = \min \left\{ -\frac{\alpha_n^{(i)}(\varepsilon)}{\lambda_n^{(i)}} \;\middle|\; i \in I_n \colon \lambda_n^{(i)} < 0 \right\}, \quad \alpha_n(\varepsilon) = \alpha_n(\varepsilon) + \nu \lambda_n,$$

and define $v_n(\varepsilon) = \sum_{i \in I_n} \alpha_n(\varepsilon) x_i$ and $\theta_n = \|v_n(\varepsilon) - w_n(\varepsilon)\|$.

- **Step 3:** Find an approximate solution γ_n of the problem

$$\min_\gamma \Big\| \sum_{j \in J_n} \gamma^{(j)} y_j - v_n(\varepsilon) \Big\|^2 \quad \text{subject to} \quad \sum_{j \in J_n} \gamma^{(j)} = 1.$$

Compute $h_n = \sum_{j \in J_n} \gamma_n^{(i)} y_j$ and $\gamma_{\min} = \min_{j \in J_n} \gamma_n^{(j)}$. If $\gamma_{\min} < 0$, compute

$$\mu = \min \left\{ \frac{\beta_n^{(j)}(\varepsilon)}{\beta_n^{(j)}(\varepsilon) - \gamma_n^{(j)}} \;\middle|\; j \in J_n \colon \gamma_n^{(j)} < 0 \right\},$$

and define $\beta_n(\varepsilon) = (1-\mu)\beta_n(\varepsilon) + \mu \gamma_n$ and

$$w_n(\varepsilon) = (1-\mu) w_n(\varepsilon) + \mu h_n, \quad \theta_n = \|v_n(\varepsilon) - w_n(\varepsilon)\|.$$

If $\gamma_{\min} = 0$, define $\beta_n(\varepsilon) = \gamma_n$, $w_n(\varepsilon) = h_n$, $\theta_n = \|h_n - w_n(\varepsilon)\|$. If $\gamma_{\min} > 0$, go to **Step 4**.

- **Step 4:** Choose any $k \in J_n$, find an approximate least-squares solution λ_n of the system

$$\sum_{j \in J_n \setminus \{k\}} \lambda^{(j)}(y_j - y_k) = 0, \qquad \sum_{j \in J_n \setminus \{k\}} \lambda^{(j)} = 1$$

and set $\lambda_n^{(k)} = -\sum_{j \in J_n \setminus \{k\}} \lambda_n^{(j)}$. Compute

$$\nu = \min \left\{ -\frac{\beta_n^{(j)}(\varepsilon)}{\lambda_n^{(j)}} \;\middle|\; j \in J_n \colon \lambda_n^{(j)} < 0 \right\}, \quad \beta_n(\varepsilon) = \beta_n(\varepsilon) + \nu \lambda_n,$$

and define $w_n(\varepsilon) = \sum_{j \in J_n} \beta_n(\varepsilon) y_j$ and $\theta_n = \|v_n(\varepsilon) - w_n(\varepsilon)\|$.

Let us present a theoretical analysis of the proposed robust meta-algorithm for computing the distance between two polytopes. Our main goal is to show that under some natural assumptions this meta-algorithm terminates after a finite number of steps and returns an approximately (in some sense) optimal solution of the problem $(\mathscr{D})$.

We start our analysis by showing that if on nth iteration of the meta-algorithm the vectors $\alpha_n(\varepsilon)$ and $\beta_n(\varepsilon)$ are such that

$$\min_{i \in I_n} \alpha_n^{(i)}(\varepsilon) = 0, \quad \min_{j \in J_n} \beta_n^{(j)}(\varepsilon) = 0 \tag{4.4}$$

(i.e. at least one of the coefficients of each of the corresponding convex combinations is zero), then on Step 2 of the meta-algorithm the approximate distance decay condition $\theta_{n+1} < \theta_n$ holds true. Therefore, the meta-algorithm increments n and moves to the next iteration without executing any other steps.

Lemma 4.9. *Suppose that* $\operatorname{diam}(P) + \operatorname{diam}(Q) > 0$*,*

$$\max \left\{ 2(\operatorname{diam}(P) + \operatorname{diam}(Q) + \operatorname{dist}(P,Q))\varepsilon, \sqrt{\frac{\operatorname{diam}(P)^2 \operatorname{diam}(Q)^2}{\operatorname{diam}(P)^2 + \operatorname{diam}(Q)^2}} \sqrt{\max\{0, 4\varepsilon\theta_0 - 2\varepsilon^2\}} \right\}$$
$$< \eta \le 2 \min \left\{ \operatorname{diam}(P)^2, \operatorname{diam}(Q)^2 \right\} \tag{4.5}$$

*and the algorithm $\mathscr{A}_\varepsilon$ with $\varepsilon \ge 0$ satisfies the following **approximate optimality condition**: for any polytopes $H = \operatorname{co}\{h_1, \dots, h_r\} \subset \mathbb{R}^d$ and $Z = \operatorname{co}\{z_1, \dots, z_s\} \subset \mathbb{R}^d$ one has*

$$\left\| \sum_{i=1}^r \alpha^{(i)} h_i - \sum_{j=1}^s \beta^{(j)} z_j \right\| < \operatorname{dist}(H, Z) + \varepsilon,$$

$$\sum_{i=1}^r \alpha^{(i)} = \sum_{j=1}^s \beta^{(j)} = 1, \quad \alpha^{(i)}, \beta^{(j)} \ge 0 \quad \forall i, j,$$

where $(\alpha, \beta) = \mathscr{A}_\varepsilon(H, Z)$. Let also for some $n \in \mathbb{N}$ the stopping criterion is not satisfied on nth iteration of Meta-algorithm 4, equalities (4.4) *hold true on Step 1, and $\theta_{k+1} < \theta_k$ for any $k \in \{0, 1, \ldots, n-1\}$. Then for θ_{n+1} computed on Step 2 one has $\theta_{n+1} < \theta_n$.*

Proof. **Part 1.** Let us first show that $\theta_k \ge \varepsilon$ for any $k \in \{0, \ldots, n\}$. Indeed, suppose by contradiction that $\theta_k < \varepsilon$ for some $k \in \{0, 1, \ldots, n\}$, that is, $\|v_k(\varepsilon) - w_k(\varepsilon)\| < \varepsilon$. Let (v_*, w_*) be an optimal solution of the problem $(\mathscr{D})$. Then obviously $\|v_* - w_*\| \le \|v_k(\varepsilon) - w_k(\varepsilon)\| < \varepsilon$, which yields

$$\|v_k(\varepsilon) - w_k(\varepsilon) - (v_* - w_*)\| \le 2\varepsilon.$$

Therefore by Proposition 4.8 for any $i \in I$ and $j \in J$ one has

$$\langle v_k(\varepsilon) - w_k(\varepsilon), x_i - v_k(\varepsilon)\rangle \ge -2\big(\operatorname{diam}(P) + \operatorname{diam}(Q) + \operatorname{dist}(P, Q)\big)\varepsilon,$$
$$\langle w_k(\varepsilon) - v_k(\varepsilon), y_j - w_k(\varepsilon)\rangle \ge -2\big(\operatorname{diam}(P) + \operatorname{diam}(Q) + \operatorname{dist}(P, Q)\big)\varepsilon.$$

Hence with the use of the first inequality in (4.5) one can conclude that the pair $(v_k(\varepsilon), w_k(\varepsilon))$ satisfies the stopping criterion of Meta-algorithm 4 (see Step 1 of the meta-algorithm), which contradicts the assumption of the lemma that the meta-algorithm performs nth iteration and the stopping criterion is not satisfied on this iteration.

Part 2. Let us now prove the statement of the lemma. By our assumption the stopping criterion is not satisfied on nth iteration. For the sake of shortness, below we will consider only the case when $\rho_{xn} < -\eta$ and $\rho_{yn} < -\eta$. The proof of the cases when either $\rho_{xn} \ge -\eta$ or $\rho_{yn} \ge -\eta$ essentially coincides with the proof of Lemma 2.14.

By condition (4.4) and the definition of indices $i_{1n} \in I_n$ and $j_{1n} \in J_n$ one has

$$\alpha_n^{(i_{1n})}(\varepsilon) = 0, \quad \beta_n^{(j_{1n})}(\varepsilon) = 0,$$

which implies that

$$v_n(\varepsilon) \in \operatorname{co}\Big\{x_i \Bigm| i \in I_n \setminus \{i_{1n}\}\Big\}, \quad w_n(\varepsilon) \in \operatorname{co}\Big\{y_j \Bigm| j \in J_n \setminus \{j_{1n}\}\Big\}.$$

Hence by the definitions of P_{n+1} and Q_{n+1} (see Step 1 of Meta-algorithm 4) one has $v_n(\varepsilon) \in P_{n+1}$ and $w_n(\varepsilon) \in Q_{n+1}$.

Define

$$x_n(t) = (1-t)v_n(\varepsilon) + t x_{i_{2n}}, \quad y_n(\tau) = (1-\tau)w_n(\varepsilon) + \tau y_{j_{2n}}.$$

Note that $x_n(t) \in P_{n+1}$ for any $t \in [0,1]$ and $y_n(\tau) \in Q_{n+1}$ for any $\tau \in [0,1]$ due to the definitions of P_{n+1} and Q_{n+1}.

For any $t \in [0,1]$ and $\tau \in [0,1]$ one has

$$\begin{aligned} f(t,\tau) := \|x_n(t) - y_n(\tau)\|^2 = \|v_n(\varepsilon) - w_n(\varepsilon)\|^2 \\ + 2t\langle v_n(\varepsilon) - w_n(\varepsilon), x_{i_{2n}} - v_n(\varepsilon)\rangle + 2\tau\langle w_n(\varepsilon) - v_n(\varepsilon), y_{j_{2n}} - w_n(\varepsilon)\rangle \\ - 2t\tau\langle v_n(\varepsilon) - x_{i_{2n}}, w_n(\varepsilon) - y_{j_{2n}}\rangle + t^2\|v_n(\varepsilon) - x_{i_{2n}}\|^2 + \tau^2\|w_n(\varepsilon) - y_{j_{2n}}\|^2. \end{aligned}$$

Estimating the inner products and the norms from above one gets

$$f(t,\tau) \le \theta_n^2 + 2t\rho_{xn} + 2\tau\rho_{yn} + 2t\tau\operatorname{diam}(P)\operatorname{diam}(Q) + t^2\operatorname{diam}(P)^2 + \tau^2\operatorname{diam}(Q)^2.$$

Let $t_* = \eta/2\operatorname{diam}(P)^2$ and $\tau_* = \eta/2\operatorname{diam}(Q)^2$. Note that $t_* \in (0,1]$ and $\tau \in (0,1]$ due to the second inequality in (4.5). Observe also that

$$f(t_*,\tau_*) \le \theta_n^2 - \frac{3\eta^2}{4\operatorname{diam}(P)^2} - \frac{3\eta^2}{4\operatorname{diam}(Q)^2} + \frac{\eta^2}{2\operatorname{diam}(P)\operatorname{diam}(Q)}$$
$$= \theta_n^2 - \frac{\eta^2}{2}\left(\frac{1}{\operatorname{diam}(P)^2} + \frac{1}{\operatorname{diam}(Q)^2}\right) - \left(\frac{\eta}{2\operatorname{diam}(P)} - \frac{\eta}{2\operatorname{diam}(Q)}\right)^2.$$

Hence applying the first inequality in (4.5) one gets that

$$\operatorname{dist}(P_{n+1},Q_{n+1})^2 \le \min_{t,\tau\in[0,1]} f(t,\tau) \le f(t_*,\tau_*)$$
$$\le \theta_n^2 - \frac{\eta^2}{2}\left(\frac{1}{\operatorname{diam}(P)^2} + \frac{1}{\operatorname{diam}(Q)^2}\right) < \theta_n^2 - 2\theta_0\varepsilon + \varepsilon^2,$$

which implies that $\operatorname{dist}(P_{n+1},Q_{n+1})^2 < (\theta_n - \varepsilon)^2$, thanks to the inequality $\theta_0 > \theta_1 > \dots \theta_n > \varepsilon$ that holds true by our assumption and the first part of the proof. Hence with the use of the approximate optimality condition on algorithm $\mathscr{A}_\varepsilon$ one has

$$\theta_{n+1} = \|v_{n+1}(\varepsilon) - w_{n+1}(\varepsilon)\| \le \operatorname{dist}(P_{n+1},Q_{n+1}) + \varepsilon < \theta_n,$$

which completes the proof. □

Remark 4.10. It should be noted that the lemma above holds true regardless of whether $\theta_0, \theta_1, \dots, \theta_n$, and $(\alpha_n(\varepsilon),\beta_n(\varepsilon))$ were computed on Step 2 or via the coefficients correction method. In particular, it holds true even if the equalities (4.4) are satisfied for $(\alpha_n(\varepsilon),\beta_n(\varepsilon))$ that was computed by the coefficient correctness method and not directly computed by the algorithm $\mathscr{A}_\varepsilon$.

With the use of the lemma above one can easily verify that if the algorithm $\mathscr{A}_\varepsilon$ is such that its output $(\alpha_n(\varepsilon),\beta_n(\varepsilon)) = \mathscr{A}_\varepsilon(P_n,Q_n)$ always satisfies equalities (4.4), then Meta-algorithm 4 never executes the coefficients correction method and terminates after a finite number of steps. The straightforward proof of this results is based on Lemma 4.9 and in essence repeats the proof of Theorem 2.17. Therefore, we omit it for the sake of shortness.

Theorem 4.11. *Let $\ell, m \ge d+1$, $\operatorname{diam}(P) + \operatorname{diam}(Q) > 0$, inequalities (4.5) hold true, and the algorithm $\mathscr{A}_\varepsilon$ with $\varepsilon > 0$ satisfy the approximate optimality condition from Lemma 4.9. Suppose also that for any polytopes $H = \operatorname{co}\{h_1,\dots,h_r\} \subset \mathbb{R}^d$ and $Z = \operatorname{co}\{z_1,\dots,z_s\} \subset \mathbb{R}^d$ with $r,s \ge d+1$ there exist $i \in \{1,\dots,r\}$ and $j \in \{1,\dots,s\}$ such that for $(\alpha,\beta) = \mathscr{A}_\varepsilon(P,Q)$ one has $\alpha^{(i)} = \beta^{(j)} = 0$. Then Meta-algorithm 4 is correctly defined, never executes Step 3 (the coefficients correction method), terminates after a finite number of iterations, and returns a pair $(v_n(\varepsilon),w_n(\varepsilon)) \in P \times Q$ such that*

$$\|v_n(\varepsilon) - w_n(\varepsilon) - (v_* - w_*)\| \le \sqrt{2\eta}, \quad \|v_n(\varepsilon) - w_n(\varepsilon)\| \le \operatorname{dist}(P,Q) + \sqrt{2\eta},$$

where (v_*,w_*) is an optimal solution of the problem $(\mathscr{D})$.

Let us finally provide sufficient conditions for the correctness and finite termination of Meta-algorithm 4 in the general case. These conditions largely coincide with the corresponding condition for Meta-algorithm 2 for finding the nearest point in a polytope.

Theorem 4.12. *Let $\ell, m \ge d+1$, $\operatorname{diam}(P) + \operatorname{diam}(Q) > 0$, inequalities (4.5) be satisfied, and the following approximate optimality conditions hold true:*

(1) for any polytopes $H = \operatorname{co}\{h_1, \ldots, h_r\} \subset \mathbb{R}^d$ and $Z = \operatorname{co}\{z_1, \ldots, z_s\} \subset \mathbb{R}^d$ one has

$$\Big\| \sum_{i=1}^{r} \alpha^{(i)} h_i - \sum_{j=1}^{s} \beta^{(j)} z_j \Big\| < \operatorname{dist}(H, Z) + \varepsilon,$$
$$\sum_{i=1}^{r} \alpha^{(i)} = \sum_{j=1}^{s} \beta^{(j)} = 1, \quad \alpha^{(i)}, \beta^{(j)} \ge 0 \quad \forall i, j,$$

where $(\alpha, \beta) = \mathscr{A}_{\varepsilon}(H, Z)$;

(2) if for some $n \in \mathbb{N}$ Meta-algorithm 4 executes Step 1 of the coefficient correction method, then $\sum_{i \in I_n} \gamma_n^{(i)} = 1$ and $\|h_n - w_n(\varepsilon)\| \le \|v_n(\varepsilon) - w_n(\varepsilon)\|$; similarly, if for some $n \in \mathbb{N}$ Meta-algorithm 4 executes Step 3 of the coefficient correction method, then $\sum_{i \in I_n} \gamma_n^{(i)} = 1$ and $\|h_n - v_n(\varepsilon)\| \le \|v_n(\varepsilon) - w_n(\varepsilon)\|$;

(3) if for some $n \in \mathbb{N}$ the vectors x_i, $i \in I_n$, are affinely independent and Meta-algorithm 4 executes Step 1 of the coefficient correction method, then $\gamma_{\min} \le 0$; similarly, if for some $n \in \mathbb{N}$ the vectors y_j, $j \in J_n$, are affinely independent and Meta-algorithm 4 executes Step 3 of the coefficient correction method, then $\gamma_{\min} \le 0$;

(4) if for some $n \in \mathbb{N}$ Meta-algorithm 4 executes Step 2 (Step 4) of the coefficients correction method, then

$$\Big\| \sum_{i \in I_n \setminus \{k\}} \gamma^{(i)} (x_i - x_k) \Big\| < \frac{\theta_{n-1} - \theta_n}{\nu}, \qquad \sum_{i \in I_n \setminus \{k\}} \gamma^{(i)} \ne 0$$
$$\left(\Big\| \sum_{j \in J_n \setminus \{k\}} \gamma^{(j)} (y_j - y_k) \Big\| < \frac{\theta_{n-1} - \theta_n}{\nu}, \qquad \sum_{j \in J_n \setminus \{k\}} \gamma^{(j)} \ne 0 \right),$$

where $\nu > 0$ is computed on Step 2 (Step 4) (if $n = 0$, then only the second inequality should be satisfied).

Then Meta-algorithm 4 is correctly defined, executes the coefficients correction method at most once per iteration, terminates after a finite number of iterations, and returns a pair $(v_n(\varepsilon), w_n(\varepsilon)) \in P \times Q$ such that

$$\|v_n(\varepsilon) - w_n(\varepsilon) - (v_* - w_*)\| \le \sqrt{2\eta}, \quad \|v_n(\varepsilon) - w_n(\varepsilon)\| \le \operatorname{dist}(P, Q) + \sqrt{2\eta},$$

where (v_*, w_*) is an optimal solution of the problem $(\mathscr{D})$.

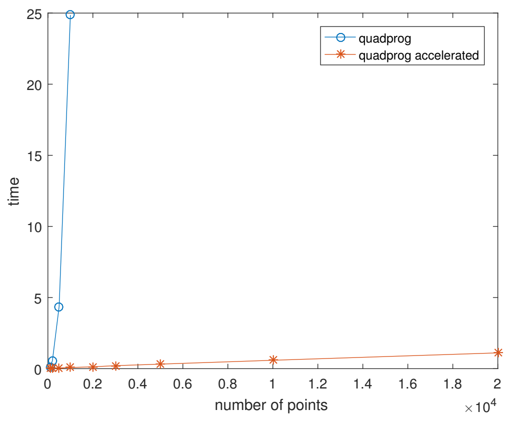
The proof of this theorem is essentially the same as the proof of Theorem 2.20. That is why we omit it for the sake of shortness.

4.3. **Numerical experiments.**

The acceleration technique for methods for computing the distance between two polytopes described in Meta-algorithm 4 was verified numerically for various values of d, ℓ, and m. Let us briefly describe the results of our numerical experiments.

We set $\ell = m$ and for each choice of d and ℓ randomly generated 10 problems. Average computation time for these problem was used to assess the efficiency of the acceleration technique.

The problem data was generated similarly to the case of the nearest point problem. First, we generated 2ℓ points $\{\widehat{x}_1, \ldots, \widehat{x}_\ell\}$ and $\{\widehat{y}_1, \ldots, \widehat{y}_\ell\}$, uniformly distributed over the d-dimensional



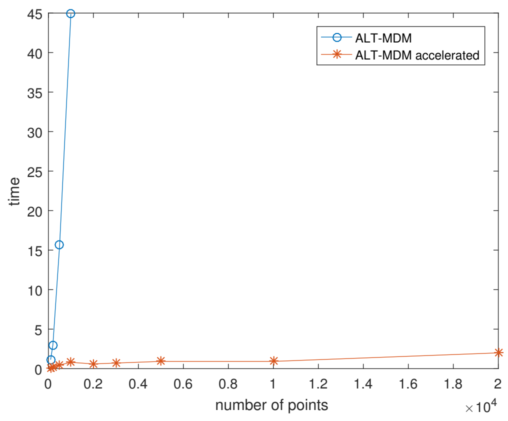


FIGURE 5. The results of numerical experiments in the case $d = 3$ for `quadprog` routine (left figure) and the ALT-MDM method (right figure).

cube $[-1,1]^d$. Then these points were compressed and shifted as follows

$$x_i = (1 + 0.01\widehat{x}_i^{(1)}, \widehat{x}_i^{(2)}, \ldots, \widehat{x}_i^{(d)}) \quad \forall i \in \{1, \ldots, \ell\},$$
$$y_j = (-1 + 0.01\widehat{y}_i^{(1)}, \widehat{y}_i^{(2)}, \ldots, \widehat{y}_i^{(d)}) \quad \forall i \in \{1, \ldots, \ell\},$$

so that the polytopes P and Q do not intersect. Numerical experiments showed that this particular problem is especially challenging for methods for computing the distance between polytopes.

Without trying to conduct comprehensive numerical experiments, we tested the acceleration technique on 2 methods: a modification of the MDM method for computing the distance between polytopes called ALT-MDM [17], and the method based on solving the quadratic programming problem

$$\min_{(\alpha,\beta)} \frac{1}{2}\Big\| \sum_{i=1}^{\ell} \alpha^{(i)} x_i - \sum_{j=1}^{\ell} \beta^{(j)} y_j \Big\|^2 \quad \text{subject to}$$
$$\sum_{i=1}^{\ell} \alpha^{(i)} = 1, \quad \alpha^{(i)} \geq 0, \quad i \in I, \qquad \sum_{j=1}^{\ell} \beta^{(j)} = 1, \quad \beta^{(j)} \geq 0, \quad j \in J$$

with the use of `quadprog`, the standard MATLAB routine for solving quadratic programming problems. We used this routine with default settings. The inequality $\Delta_x(u_k, v_k) + \Delta_y(u_k, v_k) < 10^{-4}$ was used as the termination criterion for the ALT-MDM method (see [17]). The number of iterations of this method was limited to 10^6.

Both, the ALT-MDM method and the quadratic programming method were implemented "on their own" and also incorporated within the robust acceleration technique (Meta-algorithm 4). The initial guess for the meta-algorithm was chosen as

$$I_0 = J_0 = \{1, \ldots, d+1\}, \quad P_0 = \operatorname{co}\{x_1, \ldots, x_{d+1}\}, \quad Q_0 = \operatorname{co}\{y_1, \ldots y_{d+1}\}.$$

We also set $\eta = 10^{-4}$.

The results of numerical experiments are given in Figures 5–6. Let us point out that these results are qualitatively the same as the results of numerical experiments for the nearest point

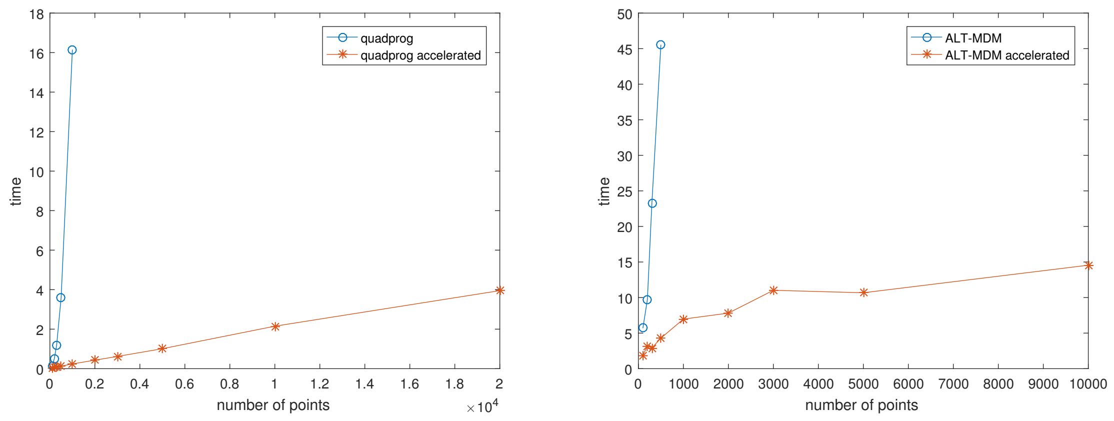


FIGURE 6. The results of numerical experiments in the case $d = 10$ for `quadprog` routine (left figure) and the ALT-MDM method (right figure).

problem given in Section 2.4. Therefore, the discussion of numerical experiments from Section 2.4 is valid for Meta-algorithm 4 as well.

Acknowledgements

The author is sincerely grateful to the anonymous reviewer for carefully reading the paper and providing valuable comments that, in particular, helped to simplify the proofs of Proposition 2.13 and Theorem 4.4. This work was performed in the IPME RAS and supported by the Russian Science Foundation (Grant No. 20-71-10032).

REFERENCES

[1] A. Bagirov, N. Karmitsa, and M. M. Mäkelä. Introduction to Nonsmooth Optimization, Springer, Cham, 2014.

[2] A. M. Bagirov, M. Gaudioso, N. Karmitsa, M. M. Mäkelä, and S. Taheri, editors, Numerical Nonsmooth Optimization, Springer, Cham, 2020.

[3] A. Barbero, J. López, and J. R. Dorronsoro, An accelerated MDM algorithm for SVM training, In: Advances in Computational Intelligence and Learning. Proceedings Of ESANN 2008 Conference, pages 421–426, 2008.

[4] D. Chakrabarty, P. Jain, and P. Kothari, Provable submodular minimization using Wolfe's algorithm, In: Proceedings of the 27th International Conference on Neural Information Processing Systems, pages 802–809, 2014.

[5] J. A. De Loera, J. Haddock, and L. Rademacher, The minimum Euclidean-norm point in a convex polytope: Wolfe's combinatorial algorithm is exponential, SIAM J. Comput. 49 (2020), 138–169.

[6] D. P. Dobkin and D. G. Kirkpatrick, Determining the separation of preprocessed polyhedra — A unified approach, In: M. S. Paterson, (ed.) Automata, Languages and Programming, ICALP 1990, pp. 400–413, Springer, Berlin, Heidelberg, 1990.

[7] H. Edelsbrunner, Computing the extreme distances between two convex polygons, J. Algorithms, 6 (1985), 213–224.

[8] S. Fujishige, Lexicographically optimal base of a polymatroid with respect to a weight vector, Math. Oper. Res. 5 (1980), 186–196.

[9] S. Fujishige and S. Isotani, A submodular function minimization algorithm based on the minimum-norm base, Pac. J. Optim. 7 (2011), 3–17.

[10] S. Fujishige and P. Zhan, A dual algorithm for finding the minimum-norm point in a polytope, J. Oper. Res. Soc. Japan 33 (1990), 188–195.

[11] S. Fujishige and P. Zhan, A dual algorithm for finding a nearest pair of points in two polytopes, J. Oper. Res. Soc. Japan, 35 (1992) 353–365.

[12] Z. R. Gabidullina, The problem of projecting the origin of Euclidean space onto the convex polyhedron, Lobachevskii J. Math. 39 (2018), 35–45.

[13] E. G. Gilbert, An iterative procedure for computing the minimum of a quadratic form on a convex set, SIAM J. Control, 4 (1966), 61–80.

[14] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, A fast procedure for computing the distance between complex objects in three-dimensional space, IEEE J. Robot. Autom. 4 (1988), 193–203.

[15] L. J. Guibas, D. Hsu, and L. Zhang, A hierarchical method for real-time distance computation among moving convex bodies, Comput. Geom. 15 (2000), 51–68.

[16] D. Kaown, A New Algorithm for Finding the Minimum Distance between Two Convex Hulls, PhD thesis, University of North Texas, 2009.

[17] D. Kaown and J. Liu, A fast geometric algorithm for finding the minimum distance between two convex hulls, In: Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, pp. 1189–1194, 2009.

[18] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K.. Murthy, A fast iterative nearest point algorithm for support vector machine classifier design, IEEE Trans. Neural Netw. 11 (2000), 124–136.

[19] B. N. Kozinets, On one algorithm for training a linear perceptron, Vychislitelnaya Tekhnika i Voprosy Programmirovaniya, 3 (1964), 80–83.
In Russian. Available at: http://oml.cmlaboratory.com/pdf/BasicKnowledge/KozinecBN_1964.pdf.

[20] S. Lacoste-Julien and M. Jaggi, An affine invariant linear convergence analysis for Frank-Wolfe algorithms, arXiv: 1312.7864, 2013.

[21] S. Lacoste-Julien and M. Jaggi, On the global linear convergence of Franke-Wolfe optimization variants, In: Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, pp. 496–504, 2015.

[22] B. Llanas, M. Fernandez de Sevilla, and V. Feliu, An iterative algorithm for finding a nearest pair of points in two convex subsets of $\mathbb{R}^n$, Comput. Math. Appl. 40 (2000), 971–983.

[23] M. E. Mavroforakis, M. Sdralis, and S. Theodoridis, A geometric nearest point algorithm for the efficient solution of the SVM classification task, IEEE Trans. Neural Netw. 18 (2007), 1545–1549.

[24] B. F. Mitchell, V. F. Dem'yanov, and V. N. Malozemov, Finding the point of a polyhedron closest to the origin, SIAM J. Control, 12 (1974), 19–26.

[25] C. M. Mückeley, Computing the vector in the convex hull of a finite set of points having minimal length, Optim. 26 (1992), 15–26.

[26] Y. Nesterov, Introductory Lectures on Convex Optimization. A Basic Course, Kluwer Academic Publishers, London, 2004.

[27] J. C. Platt, Sequential minimal optimization: A fast algorithm for training support vector machines, Technical report, Microsoft Research Technical Report MSR-TR-98-14, 1998.

[28] S. V. Plotnikov, On a scheme of algorithms for finding the distance between polytopes, Trudy Instituta Matematiki i Mekhaniki UrO RAN, 2 (1992), 214–224. In Russian.

[29] K. Sekitani and Y. Yamamoto, A recursive algorithm for finding the minimum norm point in a polytope and a pair of closest points in two polytopes, Math. Program. 61 (1993), 233–249.

[30] H. D. Sharali and G. Choi, Finding the closest point to the origin in the convex hull of a discrete set of points, Computers Oper. Res. 20 (1993), 363–370.

[31] P. I. Stetsyuk and E. A. Nurminski, Nonsmooth penalty and subgradient algorithms to solve the problem of projection onto a polytope, Cybern. Syst. Anal. 46 (2010), 51–55.

[32] P. Wolf, Finding the nearest point in a polytope, Math. Program. 11 (1976), 128–149.

[33] C. L. Yang, M. Qi, X. X. Meng, X. Q. Li, and J. Y. Wang, A new fast algorithm for computing the distance between two disjoing convex polygons based on Voronoi diagram, J. Zhejian University-Science A 7 (2006), 1522–1529.

[34] M. Zeng, Y. Yang, and J. Cheng, A generalized Mitchell-Dem'yanov-Malozemov algorithm for one-class support vector machine, Knowledge-Based Systems, 109 (2016), 17–24.

[35] G. M. Ziegle, Lectures on Polytopes, Springer Verlag, New York, 1995.