



HAL
open science

A linear optimization oracle for zonotope computation

Antoine Deza, Lionel Pournin

► **To cite this version:**

Antoine Deza, Lionel Pournin. A linear optimization oracle for zonotope computation. Computational Geometry, In press, 100, pp.101809. 10.1016/j.comgeo.2021.101809 . hal-03409685

HAL Id: hal-03409685

<https://hal.science/hal-03409685v1>

Submitted on 29 Oct 2021

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

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

A linear optimization oracle for zonotope computation

Antoine Deza

McMaster University, Hamilton, Ontario, Canada

LIX, CNRS, École Polytechnique, Palaiseau, France

Lionel Pournin*

Université Paris 13, Villetaneuse, France

Abstract

A class of counting problems asks for the number of regions of a central hyperplane arrangement. By duality, this is the same as counting the vertices of a zonotope. Efficient algorithms are known that solve this problem by computing the vertices of a zonotope from its set of generators. Here, we give an efficient algorithm, based on a linear optimization oracle, that perform the inverse task and recovers the generators of a zonotope from its set of vertices. We also provide a variation of that algorithm that allows to decide whether a polytope, given as its vertex set is a zonotope and when it is not, to compute its greatest zonotopal summand.

1. Introduction

Linear optimization consists in finding a vertex of a polyhedron that maximizes some linear functional. It is widely used in many areas of science and engineering. Linear optimization has been used for solving some prominent questions and has led to formulating a number of other widely studied problems. While linear optimization itself is known to be polynomial time solvable in the size of the problem, the complexity of simplex methods, pivot-based linear optimization algorithms, is still not known, see for instance [1] and references therein. In contrast to linear optimization that consists in finding just one vertex of a polyhedron, convex hull computation amounts to enumerate all the faces of a polytope. A number of practical algorithms have been given that address this particular problem [2, 3, 4]. Since the number of faces of a polytope of arbitrary dimension is exponential, for instance in its number of vertices, the worst case complexity of these algorithms is exponential.

This article considers a case that lies in-between linear optimization and convex hull computation. Just as with linear optimization, we are interested in the

*Corresponding author

Email addresses: `deza@mcmaster.ca` (Antoine Deza), `lionel.pournin@univ-paris13.fr` (Lionel Pournin)

small dimensional faces of polytopes (their vertices and edges) and, in keeping with convex hull computations, one of our main goals is to enumerate them. We are going to do that for a particular class of polytopes, the zonotopes or, in other words, the Minkowski sums of line segments. Recall that the Minkowski sum of two subsets A and B of \mathbb{R}^d is the set

$$A + B = \{a + b : a \in A, b \in B\}.$$

This notion is illustrated in Figure 1 in the particular case of a zonotope and in Figure 2 in general. By definition, zonotopes are more combinatorial, and can behave rather differently than arbitrary polytopes. For instance, linear optimization over a zonotope is linear time solvable in the number of its generators. Zonotopes arise in a number of counting problems related to different fields, often in terms of their dual hyperplane arrangement [5, 6, 7, 8]. These counting problems ask about the number of vertices of a zonotope, where the zonotope itself is given as the set of its generators. This can be done in practice by using an efficient, convex-hull-free algorithm that computes the vertex set of a zonotope from its set of generators, see for instance [9, 10].

Our first contribution is an efficient algorithm that performs the inverse task, also without requiring any convex hull computation. Given the vertex set of a polytope P , this algorithm will decide whether P is a zonotope and, if so, it will return its set of generators. In this sense it can also be considered a decision algorithm. It is polynomial in the number of vertices of the considered polytope. In fact, considering complexity in terms of the combined size of the input and output [2], the algorithms we propose are polynomial.

Our second contribution is an algorithm that provides a practical approach to the decomposability of a polytope in terms of Minkowski sums, another topic that has attracted significant attention [11, 12, 13, 14, 15]. It efficiently computes the *greatest zonotopal summand* of an arbitrary polytope. Let us illustrate this notion. By a summand of a polytope P , we mean any polytope Q such that P is the Minkowski sum of Q with another polytope. In Figure 2 for instance, the octagon P is the Minkowski sum of the triangle Q with the hexagon Z . In particular, Q and Z are two summands of P . Observe that Z is, up to translation, the Minkowski sum of three of its edges. In other words, Z is a zonotope. On the other hand, since Q is a triangle, no line segment—and therefore no zonotope—can be a summand of Q . In this case, Z is what we call the greatest zonotopal summand of P .

When a polytope P is given as the convex hull of a finite set of points, linear optimization is polynomial time solvable in the number of these points

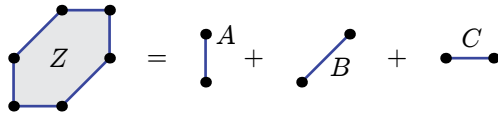


Figure 1: The zonotope Z is the Minkowski sum of the line segments A , B , and C .

since it amounts to computing the value of a linear functional at each of them. According to the theory developed in [16], deciding whether a given point is a vertex of P is then also polynomial time solvable. Similarly, deciding whether two points are the extremities of an edge of P can be done in polynomial time. Our algorithms rely on the ability to solve these two problems. In Section 2, we will give an explicit linear optimization oracle that provides a practical way to do that. In section 3, we will recall a number of properties of Minkowski sums and zonotopes, and derive further properties that will be used in the sequel.

As mentioned earlier, efficient algorithms are known that enumerate the vertices of a zonotope from its set of generators, see for instance [9, 10]. This task will appear as a subtask in our algorithms. For this reason, and in order to keep our article self-contained, we give an algorithm that performs this task in Section 4. The greatest zonotopal summand of a polytope is defined and studied at in Section 5, and the algorithm that computes it is described at the end of the section. Finally, the algorithm that enumerates the generators of a zonotope from its vertex set is given in Section 6.

2. A linear optimization oracle

We begin the section with a linear optimization oracle that allows to tell whether the convex hull of a finite subset of \mathbb{R}^d is disjoint from the affine hull of another finite subset of \mathbb{R}^d . We then show that this oracle provides a practical way to decide when a polytope is a face of another when both are given as convex hulls of finite sets of points. As a consequence, we obtain an explicit algorithm that efficiently computes the graph of a polytope given either as the set of its vertices or as the convex hull of a finite subset of \mathbb{R}^d . At the end of the section, we show how our oracle also allows to compute the extreme rays of a pointed cone given as the conic hull of a set of points.

Consider a finite subset \mathcal{A} of \mathbb{R}^d and a subset \mathcal{F} of \mathcal{A} . The convex hull of $\mathcal{A} \setminus \mathcal{F}$ and the affine hull of \mathcal{F} are non-disjoint if and only if a convex combination of $\mathcal{A} \setminus \mathcal{F}$ coincides with an affine combination of \mathcal{F} ; that is, if and only if there

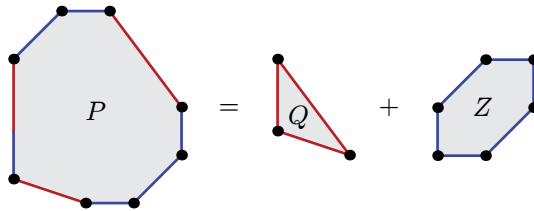


Figure 2: The octagon P is the Minkowski sum of Q and Z .

exists a family $(\alpha_a)_{a \in \mathcal{A}}$ of real numbers such that

$$\sum_{a \in \mathcal{A} \setminus \mathcal{F}} \alpha_a a - \sum_{a \in \mathcal{F}} \alpha_a a = 0, \quad (1)$$

$$\sum_{a \in \mathcal{A} \setminus \mathcal{F}} \alpha_a = 1, \quad (2)$$

$$\sum_{a \in \mathcal{F}} \alpha_a = 1, \quad (3)$$

and $\alpha_a \geq 0$ when $a \in \mathcal{A} \setminus \mathcal{F}$. In other words, checking whether $\text{conv}(\mathcal{A} \setminus \mathcal{F})$ and $\text{aff}(\mathcal{F})$ are non-disjoint amounts to finding a solution to a system of $d+2$ linear equalities (note here that (1) accounts for d of these equalities) and $|\mathcal{A}| - |\mathcal{F}|$ linear inequalities that state the non-negativity of some of the variables. This feasibility problem, which we denote by $(LO_{\mathcal{A}, \mathcal{F}})$ is polynomial time solvable in $|\mathcal{A}|$, d , and the binary size L of the input (see [16, 1]). In our case and throughout the section, L is the number of bits needed to store \mathcal{A} .

Let us illustrate how $(LO_{\mathcal{A}, \mathcal{F}})$ allows for an efficient way to compute the graph of a polytope given as the convex hull of a finite subset of \mathbb{R}^d .

Proposition 1. *Consider a finite subset \mathcal{A} of \mathbb{R}^d . A point x in \mathcal{A} is a vertex of $\text{conv}(\mathcal{A})$ if and only if it does not belong to $\text{conv}(\mathcal{A} \setminus \{x\})$.*

PROOF. The vertices of a polytope are precisely its extreme points. Therefore, a point x in \mathcal{A} is not a vertex of $\text{conv}(\mathcal{A})$ if and only if it can be written as a convex combination of \mathcal{A} whose coefficient for x is less than 1. Note that x appears on both sides of that equality. Solving this equality for x results in an equivalent equation that expresses x as a convex combination of $\mathcal{A} \setminus \{x\}$. \square

Recall that the affine hull of a single point only contains that point. Hence, according to Proposition 1, A point x in \mathcal{A} is a vertex of $\text{conv}(\mathcal{A})$ if and only if $(LO_{\mathcal{A}, \{x\}})$ is infeasible. Solving this feasibility problem for every point x in \mathcal{A} allows to recover the vertex set of $\text{conv}(\mathcal{A})$ in polynomial time in $|\mathcal{A}|$, d , and L . Proposition 1 can be generalized as follows.

Proposition 2. *Consider a finite subset \mathcal{A} of \mathbb{R}^d . The convex hull of a subset \mathcal{F} of \mathcal{A} is a face of the convex hull of \mathcal{A} if and only if the affine hull of \mathcal{F} is disjoint from the convex hull of $\mathcal{A} \setminus \mathcal{F}$.*

PROOF. Let \mathcal{F} be a subset of \mathcal{A} . Consider the orthogonal projection

$$\pi : \mathbb{R}^d \rightarrow \text{aff}(\mathcal{F})^\perp,$$

where $\text{aff}(\mathcal{F})^\perp$ the orthogonal complement of $\text{aff}(\mathcal{F})$ in \mathbb{R}^d ; that is, the set of the vectors in \mathbb{R}^d orthogonal to it. By construction, π sends all the points in \mathcal{F} to a single point x . Now observe that $\text{aff}(\mathcal{F})$ is disjoint from $\text{conv}(\mathcal{A} \setminus \mathcal{F})$ if and only if x is not contained in the convex hull of $\pi(\mathcal{A} \setminus \mathcal{F})$. By Proposition 1, this is equivalent to x being a vertex of the convex hull of $\pi(\mathcal{A})$ which, in turn, is equivalent to $\text{conv}(\mathcal{F})$ being a face of $\text{conv}(\mathcal{A})$. \square

Consider two distinct elements x and y of \mathcal{A} . According to Proposition 2, $\text{conv}(\{x, y\})$ is an edge of $\text{conv}(\mathcal{A})$ if and only if $(LO_{\mathcal{A}, \{x, y\}})$ is infeasible. Therefore, in order to compute the graph of the convex hull of a finite subset \mathcal{A} of \mathbb{R}^d , it is sufficient to solve $(LO_{\mathcal{A}, \{x\}})$ for every point x in \mathcal{A} in order to recover the vertex set \mathcal{V} of $\text{conv}(\mathcal{A})$ and then to solve $(LO_{\mathcal{A}, \{x, y\}})$ for any pair of distinct points x and y in \mathcal{V} in order to decide whether $\text{conv}(\{x, y\})$ is an edge of $\text{conv}(\mathcal{A})$. Since the number of feasibility problems to solve is quadratic in $|\mathcal{A}|$ and each of these problems is polynomial time solvable in $|\mathcal{A}|$, d , and L , we recover the following well-known result (see [17]).

Theorem 1. *Consider a finite subset \mathcal{A} of \mathbb{R}^d . The graph of $\text{conv}(\mathcal{A})$ can be computed in polynomial time in $|\mathcal{A}|$, d , and L .*

Note that, computing the whole 2-skeleton of $\text{conv}(\mathcal{A})$ with that procedure would require solving a number of feasibility problems that is exponential in $|\mathcal{A}|$. Indeed, the possible candidates for being the vertex set of a 2-dimensional face of $\text{conv}(\mathcal{A})$ would be all the sets of at least three vertices of $\text{conv}(\mathcal{A})$. Therefore, the k -skeleton of $\text{conv}(\mathcal{A})$ cannot be computed in polynomial time using the same ideas when k is greater than 1. However, for any given positive integer k , it is possible to compute in polynomial time all the faces of $\text{conv}(\mathcal{A})$ with at most k vertices and arbitrary dimension. According to Proposition 2, this amounts to solve $(LO_{\mathcal{A}, \mathcal{F}})$ for every non-empty subset \mathcal{F} of \mathcal{A} with at most k elements, whose number is at most a degree k polynomial in $|\mathcal{A}|$.

We now explain how the same ideas allow to compute in polynomial time the rays of a pointed cone spanned by a finite subset \mathcal{A} of $\mathbb{R}^d \setminus \{0\}$. This will be used in Section 4 in order to compute the vertices of a zonotope efficiently from its generators. Recall that the *cone spanned by \mathcal{A}* , or *conic hull of \mathcal{A}* , is the polyhedral cone made up of all the linear combinations with non-negative coefficients of the points in \mathcal{A} . This cone, which we denote by $\text{cone}(\mathcal{A})$, is pointed when it admits 0 as a vertex. In particular, when \mathcal{A} is made up of a single non-zero point, $\text{cone}(\mathcal{A})$ is a half-line incident to 0.

Note that, if \mathcal{A} contains a pair of linearly dependent points, then the cone spanned by \mathcal{A} is not affected if one of these points is removed from \mathcal{A} . Hence, we can assume that any two points in \mathcal{A} are linearly independent. The following proposition is the conic counterpart to Proposition 1.

Proposition 3. *Consider a finite subset \mathcal{A} of pairwise linearly independent points of $\mathbb{R}^d \setminus \{0\}$ that spans a pointed cone. The half-line spanned by a point x in \mathcal{A} is a ray of the cone spanned by \mathcal{A} if and only if the line through 0 and x is disjoint from the convex hull of $\mathcal{A} \setminus \{x\}$.*

PROOF. Consider a point x in \mathcal{A} and assume that the half-line $\text{cone}\{x\}$ is a ray of $\text{cone}(\mathcal{A})$. Let H be a supporting hyperplane of $\text{cone}(\mathcal{A})$ such that

$$\text{cone}(\mathcal{A}) \cap H = \text{cone}\{x\}.$$

Since any two points in \mathcal{A} are linearly independent, x is the only element of \mathcal{A} contained in $\text{cone}\{x\}$ and, therefore, in H . As a consequence, $\mathcal{A} \setminus \{x\}$ is

contained in one of the open half-spaces bounded by H and its convex hull is necessarily disjoint from the line through 0 and x .

Now assume that the line through 0 and x is disjoint from the convex hull of $\mathcal{A} \setminus \{x\}$. By Proposition 2, $\text{conv}\{0, x\}$ is an edge of $\text{conv}(\mathcal{A} \cup \{0\})$. Consider a supporting hyperplane H of $\text{conv}(\mathcal{A} \cup \{0\})$ such that

$$\text{conv}(\mathcal{A} \cup \{0\}) \cap H = \text{conv}\{0, x\}.$$

Since $\text{conv}(\mathcal{A} \cup \{0\}) \setminus \text{conv}\{0, x\}$ is contained in one of the open half-spaces bounded by H , and since H contains 0, then $\text{cone}(\mathcal{A}) \setminus \text{cone}(\{x\})$ is also contained in that half-space. Hence, H is a supporting hyperplane of the cone spanned by \mathcal{A} and it intersects this cone along the half-line spanned by x . In other words, that half-line is a ray of the cone spanned by \mathcal{A} . \square

Now observe that the line through 0 and a point x in \mathcal{A} is disjoint from the convex hull of $\mathcal{A} \setminus \{x\}$ if and only if $(LO_{\mathcal{A} \cup \{0\}, \{0, x\}})$ is infeasible. Again, this feasibility problem is polynomial time solvable in $|\mathcal{A}|$, d , and L . In particular, it follows from Proposition 3 that solving this problem allows to tell whether the half-line spanned by x is a ray of the conic hull of \mathcal{A} .

By these observations, we recover the following.

Theorem 2. *Consider a finite subset \mathcal{A} of pairwise linearly independent points of $\mathbb{R}^d \setminus \{0\}$ that spans a pointed cone. The rays of $\text{cone}(\mathcal{A})$ can be computed in polynomial time in $|\mathcal{A}|$, d , and L .*

The input of some of the algorithms we describe in the sequel are polytopes given as the set \mathcal{V} of their vertices. In fact, these polytopes could also be given as any finite set \mathcal{A} of points they are the convex hull of. In this case, the complexity of these algorithms would be in terms of $|\mathcal{A}|$ instead of $|\mathcal{V}|$.

3. Combinatorial properties of Minkowski additions

When P and Q are polyhedra, the faces of $P + Q$ are exactly the Minkowski sums of a face F of P and a face G of Q such that, for some non-zero vector c in \mathbb{R}^d , the linear functional $x \mapsto c \cdot x$ is maximized at F in P and at G in Q , see for instance Proposition 12.1 in [18] or Lemma 2.1 in [11]. As already mentioned, a zonotope, is the Minkowski sum of a finite set of line segments. In fact, a zonotope Z contained in \mathbb{R}^d is uniquely obtained, up to translation, as the Minkowski sum of a finite set of pairwise non-homothetic line segments incident to 0 and whose first non-zero coordinate of the other vertex is positive. We refer to these particular line segments as the *generators* of Z .

Now recall that a summand of a polytope P is a polytope Q such that P is obtained as the Minkowski sum of Q with another polytope. We recall the following decomposability characterization from [11] that we will use to compute the 1-dimensional summands of a polytope efficiently.

Lemma 1 ([11, Theorem 2.5]). *A polytope P has a summand homothetic to a polytope Q if and only if P and $P + Q$ have the same number of vertices.*

Lemma 1 is illustrated in Figure 3 in the case when P is the quadrilateral $T + e$, that admits a line segment e as a summand, and Q is a line segment f homothetic to e . As can be seen, $T + e + f$ is still a quadrilateral.

We also recall classic counting arguments that will be used to speedup the computations. If P is a polytope and s is a line segment, we denote by $\langle s \rangle_P$ the set of the edges of P homothetic to s or, equivalently, parallel to s .

Lemma 2. *Consider a d -dimensional polytope P and a line segment s , both contained in \mathbb{R}^d . If s is a summand of P , then*

- (i) *At least d edges of P are contained in $\langle s \rangle_P$,*
- (ii) *All the edges of P in $\langle s \rangle_P$ are at least as long as s ,*
- (iii) *Every shortest element of $\langle s \rangle_P$ is a summand of P .*

PROOF. Assume that there exists a polytope Q such that P is the Minkowski sum of Q and s and consider the orthogonal projection

$$\pi : \mathbb{R}^d \rightarrow \text{aff}(s)^\perp.$$

Note that the image by π of any element of $\langle s \rangle_P$ is a vertex of $\pi(P)$. In fact, π induces a bijection between $\langle s \rangle_P$ and the vertex set of $\pi(P)$. Indeed, first observe that, by convexity, π cannot send distinct elements of $\langle s \rangle_P$ to the same point. Further consider a vertex v of $\pi(P)$ and a vector c contained in $\text{aff}(s)^\perp$ such that the map $x \mapsto c \cdot x$ is maximized at v in $\pi(P)$. Observe that $\pi(P)$ coincides, up to translation, with $\pi(Q)$. Hence, $x \mapsto c \cdot x$ is maximized at a vertex of $\pi(Q)$ and, therefore, at a vertex or an edge of Q parallel to s . Denote by Q_v this vertex or edge of Q . Since c is contained in $\text{aff}(s)^\perp$, the map $x \mapsto c \cdot x$ is constant within s . Hence, according to Lemma 2.1 in [11],

$$\pi^{-1}(\{v\}) \cap P = Q_v + s. \tag{4}$$

In other words, the set of the points in P that π sends to v is an edge of P obtained as the Minkowski sum of Q_v and s . Therefore, this edge belongs to $\langle s \rangle_P$ and cannot be shorter than s . In particular Assertion (ii) holds.

Since $\pi(Q)$ is a polytope of dimension $d - 1$, it has at least d vertices. As π projects distinct segments in $\langle s \rangle_P$ to distinct points, $\langle s \rangle_P$ cannot contain less than d elements. In other words, Assertion (i) holds.

Now observe that, if Q_v is an edge of Q for every vertex v of $\pi(P)$, then Q still admits a summand homothetic to s . In this case, P has a summand

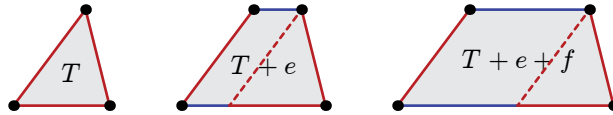


Figure 3: A triangle T , its Minkowski sum with a horizontal line segment e (center), and the Minkowski sum of $T + e$ with a line segment f homothetic to e (right).

homothetic to s that is longer than s . We can assume without loss of generality that s is a longest such summand of P . It follows that Q_v is a vertex of Q for some vertex v of $\pi(P)$. By Equality (4), some edge of P is a translate of s . According to Assertion (ii), all the shortest edges of P in $\langle s \rangle_P$ must be translates of s and, therefore, summands of P , which proves Assertion (iii). \square

In the case of zonotopes, the statement of Lemma 2 can be refined. In order to do that, we recall a well-known property of zonotopes.

Proposition 4. *A d -dimensional zonotope has at least 2^d vertices.*

PROOF. Consider a zonotope Z contained in \mathbb{R}^d . If Z is d -dimensional, then it admits d generators that are not all contained in a hyperplane of \mathbb{R}^d . The Minkowski sum of these generators is a d -dimensional combinatorial hypercube C . By construction, C is a summand of Z . Therefore, according to Lemma 2.3 from [11], there exists an injection from the vertex set of C into the vertex set of Z . Since C has 2^d vertices, the proposition follows. \square

Lemma 2 can be improved as follows in the case of zonotope.

Lemma 3. *If Z is a d -dimensional zonotope then, for every generator g of Z , $|\langle g \rangle_Z| \geq 2^{d-1}$ and all the elements of $\langle g \rangle_Z$ are translates of g .*

PROOF. Consider a d -dimensional polytope Z contained in \mathbb{R}^d and a generator g of Z . By the definition, two distinct generators of Z cannot be parallel. Since the faces of a Minkowski sum of polytopes are Minkowski sums of their faces, every edge of Z is a translate of one of its generators. As a consequence, every element of $\langle g \rangle_Z$ is necessarily a translate of g .

Consider the orthogonal projection

$$\pi : \mathbb{R}^d \rightarrow \text{aff}(g)^\perp.$$

Observe that $\pi(Z)$ is a zonotope of dimension $d - 1$ obtained, up to translation, as the Minkowski sum of the images by π of the generators of $\pi(Z)$. By the argument in the proof of Lemma 2, π induces a bijection between $\langle g \rangle_Z$ and the vertex set of $\pi(Z)$. Since $\pi(Z)$ is a $(d - 1)$ -dimensional zonotope, the desired result therefore follows from Proposition 4. \square

By Lemma 3, $|\langle g \rangle_Z|$ is at least 2^{d-1} when g is a generator of a d -dimensional zonotope Z . Note that $|\langle g \rangle_Z|$ is not necessarily a multiple of 2^{d-1} in general. For instance, the rhombic dodecahedron is a 3-dimensional zonotope whose exactly 6 edges are obtained as translates of each generator.

4. An efficient algorithm to compute zonotopes

Throughout this section, Z is a fixed zonotope contained in \mathbb{R}^d . Recall that a zonotope is, up to translation, the Minkowski sum of its generators. In this

section, we assume that Z is exactly the Minkowski sum of its generators, which can be done without loss of generality by translating Z , if needed. Denote by \mathcal{G} the set of the non-zero vertices of the generators of Z . The purpose of the section is to give an algorithm to recover the vertex set of Z from \mathcal{G} . Observe that, since Z is the Minkowski sum of its generators, it is also equal to the convex hull of all the possible subsums of \mathcal{G} ; that is,

$$Z = \text{conv} \left\{ \sum_{x \in \mathcal{X}} x : \mathcal{X} \subset \mathcal{G} \right\},$$

where, by convention, the sum of the elements of \mathcal{X} is equal to 0 when \mathcal{X} is empty. In particular, every vertex of Z is the sum of a unique subset of \mathcal{G} . However, not all the subsets of \mathcal{G} sum to a vertex of Z . The following theorem characterizes the subsets of \mathcal{G} that have this property.

Theorem 3. *Consider a subset \mathcal{X} of \mathcal{G} . The sum of the points in \mathcal{X} is a vertex of Z if and only if the set $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$ spans a pointed cone.*

PROOF. Consider the zonotope $Z_{\mathcal{X}}$ equal to the Minkowski sum of the line segments incident to 0 and whose other vertex is a point in $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$. Note that Z and $Z_{\mathcal{X}}$ are translates of one another. More precisely,

$$Z = Z_{\mathcal{X}} + x,$$

where x denotes the sum of the points in \mathcal{X} . As a consequence, x is a vertex of Z if and only if 0 is a vertex of $Z_{\mathcal{X}}$. Since $Z_{\mathcal{X}}$ is the Minkowski sum of its generators, it is equal to the convex hull of all the possible subsums of $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$. In particular $Z_{\mathcal{X}}$ admits $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$ as a subset. Hence, if 0 is a vertex of $Z_{\mathcal{X}}$, then $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$ is contained in one of the open half-spaces limited by a hyperplane through 0 and therefore spans a pointed cone.

Since $Z_{\mathcal{X}}$ is the convex hull of all the subsums of $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$, it must contain 0 and be contained in the cone spanned by $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$. Hence, if $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$ spans a pointed cone, then 0 is a vertex of $Z_{\mathcal{X}}$. \square

Let us illustrate Theorem 3 by showing that 0 is a vertex of Z . Since the first non-zero coordinate of every point in \mathcal{G} is positive, 0 is not a convex combination of \mathcal{G} . In this case, according to Proposition 1, the convex hull of $\mathcal{G} \cup \{0\}$ admits 0 as a vertex. As a consequence, the cone spanned by \mathcal{G} is pointed and, by Theorem 3, 0 is a vertex of Z .

It is worth noting that the condition in Theorem 3 can be checked efficiently. More precisely, the cone spanned by $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$ is pointed if and only if the following system of $|\mathcal{G}|$ linear inequalities is feasible.

$$c \cdot g \geq 1 \text{ for all } g \in \mathcal{X}, \tag{5}$$

$$-c \cdot g \geq 1 \text{ for all } g \in \mathcal{G} \setminus \mathcal{X}. \tag{6}$$

Indeed, the feasibility of this system is equivalent to the existence of a vector $c \in \mathbb{R}^d$ such that the map $x \mapsto c \cdot x$ is maximized exactly at 0 within the cone spanned by $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$. In other words, this cone admits 0 as a vertex.

Theorem 3 already provides a way to compute the vertices of Z . Indeed, in order to do that, it suffices to check, for each subset \mathcal{X} of \mathcal{G} whether 0 is contained in the convex hull of $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$. It is possible though, that many of these subsets do not sum to a vertex of Z . In order to avoid considering these subsets, we will use the following lemma.

Lemma 4. *Consider a subset \mathcal{X} of \mathcal{G} . If \mathcal{X} sums to a vertex x of Z , then the vertices of Z adjacent to x are the sums of x with any element of $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$ that spans an extreme ray of the conic hull of $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$.*

PROOF. Assume that the sum of the points in \mathcal{X} is a vertex x of Z . In this case, according to Theorem 3, $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$ spans a pointed cone. As in the proof of Theorem 3, we consider the zonotope $Z_{\mathcal{X}}$ whose generators are incident to 0 on one end and to a point in $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$ on the other. This zonotope is a translate of Z . More precisely,

$$Z = Z_{\mathcal{X}} + x.$$

According to this, in order to prove the lemma, we only need to show that the vertices of $Z_{\mathcal{X}}$ adjacent to 0 are exactly the points in $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$ that span an extreme ray of the conic hull of $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$.

By construction, $Z_{\mathcal{X}}$ is the Minkowski sum of its generators. Hence, $Z_{\mathcal{X}}$ is the convex hull of all the possible subsums of $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$. In particular, it is contained in the cone spanned by $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$. Now recall that the edges of a zonotope are translates of its generators. Therefore, the vertices of $Z_{\mathcal{X}}$ adjacent to 0 must be among the points from $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$. Consider a point y in $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$. The segment with vertices 0 and y is an edge of $Z_{\mathcal{X}}$ if and only if there exists a supporting hyperplane H of $Z_{\mathcal{X}}$ such that

$$Z_{\mathcal{X}} \cap H = \text{conv}\{0, y\}.$$

Since the points of $\text{cone}([- \mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}])$ are precisely the multiples by a non-negative coefficient of the points in $Z_{\mathcal{X}}$, this is equivalent to

$$\text{cone}([- \mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]) \cap H = \text{cone}\{y\}.$$

In other words, $\text{cone}\{y\}$ is an extreme ray of the cone spanned by $[-\mathcal{X}] \cup [\mathcal{G} \setminus \mathcal{X}]$. \square

According to Theorem 2, the condition in the statement of Lemma 4 can be checked efficiently using the oracle described in Section 2. As discussed above, this condition can also be checked by solving the feasibility problem made up of the inequalities (5) and (6) for each generator g of Z , where \mathcal{X} is replaced by $\mathcal{X} \cup \{g\}$ if g does not belong to \mathcal{X} and by $\mathcal{G} \setminus \{g\}$ otherwise.

Let us now give an informal description of our algorithm that computes the vertices of Z from \mathcal{G} . Recall that Z admits 0 as a vertex. Our algorithm starts from that vertex and computes all the vertices of Z adjacent to it. According to Lemmas 2 and 4, this can be done in polynomial time in $|\mathcal{G}|$, d , and the binary

size L required to store \mathcal{G} . Then the procedure is repeated and computes the vertices of Z adjacent to the new vertices of Z that have been discovered, and so on until the neighbors of all the discovered vertices have been computed. Since the graph of a polytope (made up of its vertices and edges) is connected, this indeed computes all the vertices of Z . In order to further speedup our algorithm, we use the following proposition that allows to compute only a subset of the neighbors of each vertex.

Proposition 5. *If \mathcal{X} is a non-empty subset of \mathcal{G} that sums to a vertex x of Z , then there exists a vertex y of Z adjacent to x such that $x - y \in \mathcal{X}$.*

PROOF. The proof proceeds by induction on the dimension of Z . If Z has dimension 0, then \mathcal{G} is empty and the desired result immediately holds. Assume that the dimension of Z is positive, and that the desired statement holds for any zonotope of dimension less than the dimension of Z .

Consider a non-empty subset \mathcal{X} of \mathcal{G} that sums to a vertex x of Z . Since \mathcal{X} is non-empty and the first non-zero coordinate of any point it contains is positive, then x is necessarily distinct from 0 and its first non-zero coordinate must be positive. We will review two cases.

Assume that the first coordinate of x is positive. As Z contains 0, Z must have an edge incident to x whose first coordinate of the other vertex is less than that of x . By Lemma 4, the other vertex y of this edge is such that either $x - y$ belongs to \mathcal{X} or to $-\mathcal{G} \setminus \mathcal{X}$. Since the first coordinate of the points in \mathcal{G} is non-negative, $x - y$ must belong to \mathcal{X} , as desired.

Now assume that the first coordinate of x is equal to 0. Since the first coordinate of the points in \mathcal{G} is non-negative, it follows that the first coordinate of all the points in \mathcal{X} must be equal to 0. Moreover, the zonotope generated by the elements of \mathcal{G} whose first coordinate is equal to 0 is a proper face of Z . Hence, the proposition holds by induction. \square

Consider a vertex x of Z and the subset \mathcal{X} of \mathcal{G} it is the sum of. A consequence of Proposition 5 is that x can be reached from 0 by a path in the graph of Z that visits only vertices equal to subsums of \mathcal{X} . In other words, in order to discover new vertices of Z from x in the algorithm sketched above, one only needs to check the points y such that $y - x$ belongs to $\mathcal{G} \setminus \mathcal{X}$, and the algorithm will still compute all the vertices of Z . This is what Algorithm 1 does.

Let us give a detailed description of Algorithm 1 that explores the graph of the zonotope as for example reverse-search does [3]. In this algorithm, \mathcal{T} is the set of the vertices of Z that have been discovered, but not treated yet in the sense that their neighbors in the graph of Z are still to be computed. The set of the vertices that have been treated, in the same sense is denoted by \mathcal{V} . Initially, \mathcal{T} only contains 0 and \mathcal{V} is empty. Upon completion of the algorithm, \mathcal{V} is the set of the vertices of Z . For each point x in $\mathcal{T} \cup \mathcal{V}$, the subset of \mathcal{G} that sums to x is denoted by $\xi(x)$. For instance, $\xi(0)$ is equal to \emptyset .

While \mathcal{T} is non-empty, the algorithm picks a point x from \mathcal{T} , and considers all the points y that are the sum of x with an element of $\mathcal{G} \setminus \xi(x)$. By Proposition 5,

Algorithm 1: Computing \mathcal{V} from \mathcal{G}

```
1  $\mathcal{T} \leftarrow \{0\}$ 
2  $\xi(0) \leftarrow \emptyset$ 
3  $\mathcal{V} \leftarrow \emptyset$ 
4 while  $\mathcal{T} \neq \emptyset$  do
5     Pick an element  $x$  of  $\mathcal{T}$ 
6     for every point  $y$  in  $x + \mathcal{G} \setminus \xi(x)$  do
7         if  $y$  does not belong to  $\mathcal{T} \cup \mathcal{V}$  then
8             if  $y - x$  spans a ray of  $\text{cone}([- \xi(x)] \cup [\mathcal{G} \setminus \xi(x)])$  then
9                  $\mathcal{T} \leftarrow \mathcal{T} \cup \{y\}$  and  $\xi(y) \leftarrow \xi(x) \cup \{y - x\}$ 
10            end
11        end
12    end
13     $\mathcal{T} \leftarrow \mathcal{T} \setminus \{x\}$  and  $\mathcal{V} \leftarrow \mathcal{V} \cup \{x\}$ 
14 end
15 Return  $\mathcal{V}$ 
```

one can restrict to only consider these points to enumerate the vertex set of Z . In Line 7, the algorithm first checks whether y has not been discovered yet (which can be done in logarithmic time in the number of vertices of Z using an appropriate data structure). If y has not been discovered, the algorithm checks in Line 9 whether y is a vertex of Z , using the condition stated by Lemma 4. According to Theorem 2, this can be done in polynomial time in $|\mathcal{G}|$, d , and the binary size L required to store \mathcal{G} . If y is a vertex of Z , then it is inserted in \mathcal{T} and $\xi(y)$ is computed in Line 10. Once x has been treated, it is removed from \mathcal{T} and placed in \mathcal{V} in Line 14.

Theorem 4. *There exists a polynomial function $p : \mathbb{R}^3 \rightarrow \mathbb{R}$ such that the vertex set of a d -dimensional zonotope with n vertices and m generators can be computed from the set of its generators in time $O(np(m, d, L))$, where L is the number of bits required to store all these generators.*

PROOF. By Theorem 2, there exists a polynomial function $q : \mathbb{R}^3 \rightarrow \mathbb{R}$ such that the test in Line 9 of Algorithm 1 can be done in time $O(q(m, d, L))$ for a d -dimensional zonotope with m generators, where L is the binary size required to store these generators. Hence, according to the description of the algorithm, the vertex set of a d -dimensional zonotope with n vertices and m generators can be computed from the set of its generators in time

$$O(nm[q(m, d, L) + \log n]).$$

Since n is at most 2^m , we obtain the desired result. \square

5. The greatest zonotopal summand of a polytope

We introduce the greatest zonotopal summand of a polytope in this section. We also discuss some of its properties and give an efficient algorithm to compute it for a polytope given as the set of its vertices. In the remainder of the section P is a fixed d -dimensional polytope with n vertices.

Denote by \mathcal{E} the set made up of the edges of P that are also summands of P and consider a segment e in \mathcal{E} . We refer to as e^* the unique translate of e whose one vertex is incident to 0 and whose first non-zero coordinate of the other vertex is positive. According to Lemma 2, any edge f of P in the intersection $\mathcal{E} \cap \langle e \rangle_P$ has the same length as e and, therefore, e^* and f^* must coincide. In the remainder of the section, we consider the set

$$\mathcal{G} = \{e^* : e \in \mathcal{E}\}.$$

While in Section 4, \mathcal{G} was a set of points, here \mathcal{G} contains line segments. However, in both cases, \mathcal{G} describes the generators of a zonotope.

Definition 1. We call *greatest zonotopal summand of P* and denote by $z(P)$ the Minkowski sum of the line segments contained in \mathcal{G} .

By this definition, $z(P)$ is a zonotope. Let us show that $z(P)$ is indeed a summand of P and that it is the greatest such summand.

Theorem 5. *For any polytope P , there exists a polytope $r(P)$ with no 1-dimensional summand such that the Minkowski sum $z(P) + r(P)$ is equal to P .*

PROOF. Note that distinct line segments in \mathcal{G} cannot be parallel. Hence, none of these segments admit a summand homothetic to another. Since each line segment in \mathcal{G} is a summand of P , their Minkowski sum is necessarily a summand of P . Therefore, there exists a polytope $r(P)$ such that

$$P = z(P) + r(P). \tag{7}$$

Now assume that $r(P)$ has a 1-dimensional summand s . In this case, s is also a summand of P and, according to Lemma 2, so are the shortest elements of $\langle s \rangle_P$. Let e be a shortest element of $\langle s \rangle_P$. By construction, e^* must belong to \mathcal{G} and is therefore a generator of $z(P)$. According to (7), $e^* + s$ is then a summand of P . However, by Lemma 2, $e^* + s$ cannot be longer than e , proving that s cannot be a summand of $r(P)$ in the first place. \square

Corollary 1. *If P admits a zonotope Z as a summand, then $z(P)$ necessarily also admits Z as a summand.*

PROOF. By Theorem 5,

$$P = z(P) + r(P),$$

where $r(P)$ does not admit a 1-dimensional summand. Therefore, if a zonotope is a summand of P , then all of its generators must be summands of $z(P)$. Hence, that zonotope is itself a summand of $z(P)$. \square

Recall that n denotes the number of vertices of P . The edges of P are quadratically-many in n and, as shown in Section 2 they can all be computed in polynomial time in n , d , and the binary size L required to store all the vertices of P . It turns out that the vertex set of the Minkowski sum of P with a line segment can also be computed in polynomial time in n , d , and L . In fact, we have the following more general observation.

Remark 1. Consider two finite subsets \mathcal{A} and \mathcal{B} of \mathbb{R}^d . According to Theorem 1, the graph of the Minkowski sum $\text{conv}(\mathcal{A}) + \text{conv}(\mathcal{B})$ can be computed in polynomial time in $|\mathcal{A}||\mathcal{B}|$, d , and the number of bits required to store the points in \mathcal{A} and in \mathcal{B} . Indeed, this amounts to compute the graph of the convex hull of $\mathcal{A} + \mathcal{B}$, a subset of at most $|\mathcal{A}||\mathcal{B}|$ points of \mathbb{R}^d .

As a consequence of this remark, the set \mathcal{G} of the generators of $z(P)$ can be computed in polynomial time in n , d , and L as well. Algorithm 2 is a polynomial

Algorithm 2: Computing \mathcal{G} and \mathcal{W} from \mathcal{V}

```

1  $\mathcal{R} \leftarrow \emptyset$ 
2 for every subset  $\{x, y\}$  of  $\mathcal{V}$  such that  $x \neq y$  do
3    $E \leftarrow \text{conv}\{x, y\}$ 
4   if  $e$  is an edge of  $P$  then
5     if some segment  $s$  in  $\mathcal{R}$  is parallel to  $e$  then
6        $\mu(s) \leftarrow \mu(s) + 1$ 
7       if  $e$  is shorter than  $s$  then
8          $s \leftarrow e^*$ 
9       end
10    else
11       $\mathcal{R} \leftarrow \mathcal{R} \cup \{e^*\}$  and  $\mu(e^*) \leftarrow 1$ 
12    end
13  end
14 end
15  $\mathcal{W} \leftarrow \mathcal{V}$  and  $\mathcal{G} \leftarrow \emptyset$ 
16 for every segment  $s$  in  $\mathcal{R}$  do
17   if  $\mu(s) \geq d$  then
18     Compute the vertex set  $\mathcal{X}$  of  $\text{conv}(\mathcal{W}) + s$ 
19     if  $|\mathcal{X}| = |\mathcal{W}|$  then
20        $\mathcal{G} \leftarrow \mathcal{G} \cup \{s\}$ 
21       for every point  $x$  in  $\mathcal{W} \setminus \mathcal{X}$  do
22         Replace  $x$  in  $\mathcal{W}$  by  $x - s^+$ 
23       end
24     end
25  end
26 end
27 Return  $\mathcal{G}$  and  $\mathcal{W}$ 

```

time algorithm in n , d , and L that not only computes \mathcal{G} , but also the vertex set of $r(P)$. This vertex set is denoted by \mathcal{W} in the algorithm. The vertex set of P , denoted by \mathcal{V} , is the only input of the algorithm. Algorithm 2 is split in two parts. The first part, from Line 1 to Line 14 computes a set \mathcal{R} of candidates for membership in \mathcal{G} . In other words, \mathcal{R} admits \mathcal{G} as a subset. More precisely, \mathcal{R} is obtained by selecting and then translating edges of P such that no two of them are parallel. In addition, any such selected edge e is shortest in $\langle e \rangle_P$. Note that the translation takes place in Lines 8 and 11 where e^* is stored in \mathcal{R} instead of e . In this first part of the algorithm, a map $\mu : \mathcal{R} \rightarrow \mathbb{N}$ is also computed in Lines 6 and 11 such that $\mu(s) = |\langle s \rangle_P|$ for every segment s in \mathcal{R} .

In the second part of algorithm 2, from Line 15 to Line 26, \mathcal{W} is initially set equal to \mathcal{V} and \mathcal{G} to the empty set. The segments in \mathcal{R} that are summands of P are placed in \mathcal{G} in Line 20 and subtracted from \mathcal{W} by the loop in Lines 21 to 23. Lines 18 and 19 check whether a segment s in \mathcal{R} is a summand of P using the Minkowski sum of s with the convex hull of \mathcal{W} instead of its Minkowski sum with P , allowing for some speedup. This is valid because during the execution of the loop at Line 16, the 1-dimensional summands of P remain summands of the convex hull of \mathcal{W} until they are found and subtracted from \mathcal{W} .

Let us explain how the subtraction, in Lines 21 to 23 of Algorithm 2, works. By construction, 0 is a vertex of every segment in \mathcal{R} . For any segment s in \mathcal{R} , let s^+ stand for the non-zero vertex of s . If the convex hull of \mathcal{W} admits s as a summand; that is, if it coincides with $R + s$ for some polytope R , then \mathcal{W} is naturally partitioned into the points that are vertices of R (because they are the Minkowski sum of a vertex of R with 0) and the points equal to the sum of a vertex of R with s^+ . The latter subset is precisely made up of the points in \mathcal{W} that are further displaced by another s^+ when the vertex set \mathcal{X} of $\text{conv}(\mathcal{W}) + s^+$ is computed in Line 18. Therefore, in order to recover the vertex set of R , one only needs to subtract s^+ from any point in $\mathcal{W} \setminus \mathcal{X}$, which is done in Line 22, and to keep all the other points in \mathcal{W} .

As explained above, all the computations carried out by Algorithm 1 are polynomial and they are carried out at most a quadratic number of times. In addition, we have seen in Section 4 that the vertex set of a zonotope can also be computed from its generators in polynomial time.

We therefore obtain the following.

Theorem 6. *For any polytope P , the vertex sets of $z(P)$ and $r(P)$ can be computed in polynomial time in n , d , and the binary size required to store the vertices of P .*

6. Deciding whether a polytope is a zonotope

Throughout this section, P is a fixed d -dimensional polytope with n vertices, just as in Section 5. Observe that, when P is a zonotope, $z(P)$ is a translate of P and $r(P)$ shrinks to a single point. In particular, Algorithm 2 allows to decide whether a polytope P is a zonotope: this will be the case when the set \mathcal{W} of the vertices of $r(P)$ returned by this algorithm is made up of a single point. In

Algorithm 3: Deciding whether P is a zonotope

```
1  $\mathcal{G} \leftarrow \emptyset$ 
2 for every subset  $\{x, y\}$  of  $\mathcal{V}$  such that  $x \neq y$  do
3    $e \leftarrow \text{conv}\{x, y\}$ 
4   if  $e$  is an edge of  $P$  then
5     if some segment  $s$  in  $\mathcal{G}$  is parallel to  $E$  then
6       if  $e$  and  $s$  have different lengths then
7          $\text{Return } 0$ 
8       end
9        $\mu(s) \leftarrow \mu(s) + 1$ 
10    else
11       $\mathcal{G} \leftarrow \mathcal{G} \cup \{e^*\}$  and  $\mu(e^*) \leftarrow 1$ 
12    end
13  end
14 end
15 for every segment  $s$  in  $\mathcal{G}$  do
16   if  $\mu_P(s) < 2^{d-1}$  then
17      $\text{Return } 0$ 
18   end
19 end
20 for every segment  $s$  in  $\mathcal{G}$  do
21   Compute the vertex set of  $P + s$ 
22   if the number of vertices of  $P$  and  $P + s$  is different then
23      $\text{Return } 0$ 
24   end
25 end
26  $\text{Return } \mathcal{G}$ 
```

order to solve this decision problem, we can give an alternative algorithm that terminates faster in case the polytope is not a zonotope. Note that, however, the worst case complexity the same as that of Algorithm 2.

Just as Algorithm 2, Algorithm 3 takes as its only input the vertex set \mathcal{V} of P . The algorithm returns 0 when P is not a zonotope. When P is a zonotope, it coincides, up to translation, with $z(P)$. In this case, Algorithm 3 returns the set \mathcal{G} of the generators of $z(P)$. This algorithm consists in three parts. The first part, from Line 1 to Line 14 computes a set of line segments that are candidate generators of P . Note that this set of line segments is already denoted \mathcal{G} since, if P turns out to be a zonotope, then this set is precisely the set of the generators of $z(P)$. The computation of \mathcal{G} in Algorithm 3 is very similar to the computation of \mathcal{R} in Algorithm 2, except that the algorithm terminates in Line 7 if it finds two parallel edges of P of different lengths. The map $\mu : \mathcal{G} \rightarrow \mathbb{N}$ such that, for any segment s in \mathcal{G} , $\mu(s) = |\langle s \rangle_P|$ is computed in Lines 9 and 11 like the map $\mu : \mathcal{R} \rightarrow \mathbb{N}$ is in Algorithm 2.

In the second part of Algorithm 3, from Line 15 to Line 19, every line segment in \mathcal{G} is checked, and the algorithm terminates in Line 17 if, for such a segment s , $\mu(s) < 2^{d-1}$. Indeed according to Lemma 3, P cannot be a zonotope in this case. In the third part of the algorithm, from Line 20 to Line 25, the vertex sets of the Minkowski sums of P with the line segments in \mathcal{G} are computed, and the algorithm terminates in Line 23 if for such a segment s , P and $P + s$ do not have the same number of vertices.

Observe that, if Algorithm 3 does not return 0 then, for every edge e of P , all the segments in $\langle e \rangle_P$ are translates of e . Moreover, in this case every edge of P is a summand of P because any translate of a summand of P remains a summand of P . Therefore, the set \mathcal{G} returned by Algorithm 3 is indeed the set of the generators of $z(P)$. As a consequence, P is a zonotope.

Finally, observe that a zonotope is always centrally-symmetric and, therefore, has an even number of vertices. This very simple test can be done at the beginning of the algorithm to allow for some further speedup.

Acknowledgement. The authors thank the anonymous referees for providing valuable comments and suggestions.

References

- [1] T. Illés, T. Terlaky, Pivot versus interior point methods: pros and cons, *European Journal of Operational Research* 140 (2002) 170–190.
- [2] D. Avis, D. Bremner, R. Seidel, How good are convex hull algorithms?, *Computational Geometry* 7 (1997) 265–301.
- [3] D. Avis, K. Fukuda, A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra, *Discrete & Computational Geometry* 8 (1992) 295–313.
- [4] F. Preparata, M. Shamos, *Computational geometry: an introduction*, Springer, 1985.
- [5] J. Block, S. Weinberger, Aperiodic tilings, positive scalar curvature, and amenability of spaces, *Journal of the American Mathematical Society* 5 (1992) 907–918.
- [6] A. Deza, G. Manoussakis, S. Onn, Primitive zonotopes, *Discrete & Computational Geometry* 60 (2018) 27–39.
- [7] S. C. Gutekunst, K. Mészáros, T. K. Petersen, Root cones and the resonance arrangement, *Electronic Journal of Combinatorics* 28 (2021) P1.12.
- [8] M. Melamed, S. Onn, Convex integer optimization by constantly many linear counterparts, *Linear Algebra and its Applications* 447 (2014) 88–109.

- [9] D. Avis, K. Fukuda, Reverse search for enumeration, *Discrete Applied Mathematics* 65 (1992) 21–46.
- [10] H. Edelsbrunner, J. O’Rourke, R. Seidel, Constructing arrangements of lines and hyperplanes with applications, *SIAM Journal on Computing* 15 (1986) 341–363.
- [11] A. Deza, L. Pournin, Diameter, decomposability, and Minkowski sums of polytopes, *Canadian Mathematical Bulletin* 62 (2019) 741–755.
- [12] M. Kallay, Decomposability of polytopes, *Israel Journal of Mathematics* 41 (1982) 235–243.
- [13] W. Meyer, Indecomposable polytopes, *Transactions of the American Mathematical Society* 190 (1974) 77–86.
- [14] K. Przesławski, D. Yost, Decomposability of polytopes, *Discrete & Computational Geometry* 39 (2008) 460–468.
- [15] G. C. Shephard, Decomposable convex polyhedra, *Mathematika* 10 (1963) 89–95.
- [16] M. Grötschel, L. Lovász, A. Schrijver, Geometric algorithms and combinatorial optimization, Vol. 2 of *Algorithms and Combinatorics*, Springer, 1993.
- [17] K. Clarkson, More output-sensitive geometric algorithms, in: *FOCS 1994 (35th Annual IEEE Symposium on Foundations of Computer Science)*, 1994, pp. 695–702.
- [18] K. Fukuda, *Polyhedral Computation*, ETH research collection, Zürich, Switzerland, 2020, <https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/426218/PolyCompBook20200829.pdf>.