On the co-NP-Completeness of the Zonotope Containment Problem

Adrian Kulmburg*, Matthias Althoff

Chair of Robotics, Artificial Intelligence and Real-time Systems
Technische Universität München, Boltzmannstr. 3, 85748 Garching b. München, Germany.

Abstract

We introduce a new type of norm for non-degenerate zonotopes to solve the point containment problem, i.e., whether a point lies in a zonotope. With this norm we prove the co-NP-completeness of the zonotope containment problem, i.e., whether a zonotope is contained within another one. We propose novel algorithms to solve the zonotope containment problem exactly in polynomial time when fixing the dimension or the number of generators of either of the two zonotopes. In addition, we propose an optimisation-based algorithm, that is particularly suitable for disproving containment for zonotopes.

Keywords: zonotope, containment problem, zonotope norm, computational complexity, computational geometry, optimization.

1. Introduction

For two sets U and V, deciding whether $U \subseteq V$ defines a class of decision problems called containment problems. Although in general these problems cannot be solved algorithmically, containment problems are often solvable if some structure for U and V is assumed, e.g., when U and V are polytopes or zonotopes. Zonotopes are sets that cannot only be represented compactly but are also closed under linear maps and Minkowski sum. Owing to these favourable properties, zonotopes are used for reachability analysis [1], set-based observers [2], fault detection [3], robust control [4], controller synthesis [5], and conformance checking [6]. The aforementioned applications often require solving the zonotope containment problem, i.e., whether a zonotope is contained in another one. This is, for instance, useful for verifying an invariant of a discrete-time system by checking whether the reachable set of the next step is contained in the previous one.

Previous research has primarily focused on the containment problem of the more general class of polytopes [7], [8]. In [7], it is shown that the complexity of the problem depends heavily on the form of the input, as polytopes may be represented using either a halfspace or a vertex representation. The work in [9] attempted to solve the containment problem by proposing a necessary but not sufficient condition for $U \subseteq V$ to hold if U and V are zonotopes. This condition can be determined in polynomial time, and numerical results show that it is correct in a large number of cases, but the question of whether this could be extended to a sufficient criterion running in polynomial time was left unanswered.

Contributions. In this paper, we show that an exact algorithm that solves the zonotope containment problem in polynomial time does not exist unless P = NP, disproving the conjecture left unanswered in [9]. To do so, we will first define a norm for non-degenerate zonotopes. This enables us to transform the containment problem into an optimisation problem, which will be shown to be co-NP-complete. Finally, we propose several algorithms[1] that solve the zonotope containment prob-

*Corresponding author

Email addresses: adrian.kulmburg@tum.de (Adrian Kulmburg), althoff@in.tum.de (Matthias Althoff)

[1]The scripts of our results, including the code generating the figures in this document, are available at the URL https://github.com/AdrianKulmburg/ZonotopeContainmentProblem
The implementation depends on the CORA toolbox [10]. The algorithms presented in section 4 will also be made available for the CORA 2021 release.

lem in polynomial time for certain fixed parameters, as well as one optimisation-based algorithm that solves the problem efficiently in practice.

2. Preliminaries

2.1. Notation

We denote vectors by lower-case letters with an arrow, e.g., $\vec{v}$, matrices by underlined, upper-case letters, e.g., $\underline{M}$, v_i is the i-th coordinate of $\vec{v}$, and M_{ij} the (i,j)-th coordinate of $\underline{M}$. $\vec{0}_n$ and $\vec{1}_n$ are n-dimensional vectors of zeros and ones, respectively, $\underline{0}_{n\times m}$ is the $n \times m$ zero matrix, and $\underline{I}_{n\times n}$ is the n-dimensional identity matrix. $\underline{M}^T$ or $\vec{v}^T$ indicates the transpose and $\vec{v} \leq \vec{w} \Leftrightarrow v_i \leq w_i \ \forall i = 1, ..., n$. U° is the topological interior of U and ∂U the topological boundary of U. $U \backslash V$ denotes the set of elements in U that are not in V. For points $\vec{v}_1, ..., \vec{v}_k \in \mathbb{R}^n$, $\mathrm{conv}(\vec{v}_1, ..., \vec{v}_k)$ is the convex hull of these points. For a polytope $P \subset \mathbb{R}^n$, we write its dual polytope as P^Δ [11, Definition 2.10., p.61]. We always assume that P^Δ is embedded in $\mathbb{R}^n$ via the canonical isomorphism between $\mathbb{R}^n$ and its dual vector space $(\mathbb{R}^n)^*$, so that we may write $P^\Delta \subseteq \mathbb{R}^n$.

Decision and optimization problems are denoted by upper-case letters, e.g., ZC. Algorithms are written using the verbatim font, e.g., `opt`.

2.2. Set Representations

Let us, first, review some basic set representations. In the following, n is some arbitrary number in $\mathbb{N}$.

Definition 1. A *polytope* is a set $P \subseteq \mathbb{R}^n$, that can either be represented as

$$P = \left\{ \vec{x} \in \mathbb{R}^n \middle| \underline{H}\vec{x} \leq \vec{h} \right\}, \tag{1}$$

for some matrix $\underline{H} \in \mathbb{R}^{s \times n}$, a vector $\vec{h} \in \mathbb{R}^s$, and $s, m \in \mathbb{N}$, or that can represented as

$$P = \mathrm{conv}(\vec{v}_1, ..., \vec{v}_k), \tag{2}$$

for some points $\vec{v}_1, ..., \vec{v}_k \in \mathbb{R}^n$, and $k \in \mathbb{N}$. Both representations are equivalent, i.e. one can transform one representation to the other and vice-versa [11, Theorem 1.1, p. 29]. We call the matrix $\underline{H}$ the *halfspace matrix*, $\vec{h}$ the *halfspace coefficients*, and $\vec{v}_1, ..., \vec{v}_k$ the *vertices* of P. The representations (1) and (2) of the polytope P are called the H-representation and the V-representation, respectively.

Definition 2. A *zonotope* is a set $Z \subset \mathbb{R}^n$ for which there exists some $m \in \mathbb{N}$, a matrix $\underline{G} \in \mathbb{R}^{n\times m}$, and a vector $\vec{c} \in \mathbb{R}^n$ such that

$$Z = \langle \vec{c}, \underline{G} \rangle := \left\{ \vec{c} + \underline{G}\vec{\beta} \middle| \vec{\beta} \in [-1,1]^m \right\}. \tag{3}$$

The vector $\vec{c}$ is called the centre, matrix $\underline{G}$ is called the generator matrix, and column vectors $\vec{g}_1, ..., \vec{g}_m$ of $\underline{G}$ are called the generators.

A zonotope Z is also a polytope, with halfspace representation $(\underline{H}, \vec{h})$, with $\underline{H} \in \mathbb{R}^{s\times n}$ for some $s \leq f$ [12, p. 238-239], where f is the number of facets of the zonotope. A zonotope with m generators has at most $f := 2\binom{m}{n-1}$ facets. This bound is tight, i.e., for any $m, n \in \mathbb{N}$ there exists a zonotope with that number of facets. We denote by `halfspace`(Z) the operation that computes $(\underline{H}, \vec{h})$ for a zonotope Z, and by `vertices`(Z) the operation that computes the vertices of Z [10, p. 128].

For later use, we define a norm that can be defined for any bounded polytope P:

Definition 3. Let $P \subseteq \mathbb{R}^n$ be a bounded polytope with halfspace matrix $\underline{H} \in \mathbb{R}^{s\times n}$ and coefficients $\vec{h} \in \mathbb{R}^s$ satisfying $h_i \neq 0$ for $i = 1, ..., s$. Let $\vec{\eta}_1, ..., \vec{\eta}_s$ be the row vectors of $\underline{H}$. We call the function

$$\begin{aligned} \mathcal{S}_P :& \mathbb{R}^n \to [0, \infty) \\ & \vec{x} \mapsto \max\left\{0, \frac{\vec{\eta}_1^T\vec{x}}{h_1}, ..., \frac{\vec{\eta}_s^T\vec{x}}{h_s}\right\} \end{aligned} \tag{4}$$

the polyhedral asymmetric norm of P.

Lemma 1. *The function $\mathcal{S}_P$ is an asymmetric norm [13], [14], i.e., there holds:*

1. *Triangle inequality: $\mathcal{S}_P(\vec{x}+\vec{y}) \leq \mathcal{S}_P(\vec{x})+\mathcal{S}_P(\vec{y})$ for $\vec{x}, \vec{y} \in \mathbb{R}^n$.*
2. *For $\vec{x} \in \mathbb{R}^n$ and $a > 0$, $\mathcal{S}_P(a\vec{x}) = a\mathcal{S}_P(\vec{x})$.*

If P is symmetric such that $\mathcal{S}_P(\vec{x}) = \mathcal{S}_P(-\vec{x})$, the function $\mathcal{S}_P$ becomes an actual norm. Furthermore, the unit ball of $\mathcal{S}_P$, i.e., the set of vectors $\vec{x} \in \mathbb{R}^n$ such that $\mathcal{S}_P(\vec{x}) \leq 1$, corresponds to the polytope P.

3. The Zonotope Containment Problem

3.1. Solving the Point Containment Problem for Zonotopes

We now turn our attention to the problem of checking if a point $\vec{p} \in \mathbb{R}^n$ lies in a zonotope

$Z = \langle \vec{c}, \underline{G} \rangle \subset \mathbb{R}^n$, with $\underline{G} \in \mathbb{R}^{n\times m}$. By the definition in (3), it suffices to check whether there exists a vector $\vec{\gamma} \in [-1,1]^m$ such that

$$\vec{p} = \vec{c} + \underline{G}\vec{\gamma}. \tag{5}$$

This is equivalent to checking whether

$$1 \geq \nu(\vec{p}) := \min_{\vec{\gamma}\in\mathbb{R}^m} \|\gamma\|_\infty, \text{ subject to } \underline{G}\vec{\gamma} = \vec{p} - \vec{c}. \tag{6}$$

To solve this, we can introduce new variables and constraints to transform (6) into a linear optimisation problem that runs in polynomial time [15]. Specifically, we introduce an additional variable ω, on which we impose the following constraints:

$$\begin{aligned} \omega &\geq \gamma_i \quad \text{for } i = 1, ..., m, \\ \omega &\geq -\gamma_i \quad \text{for } i = 1, ..., m. \end{aligned} \tag{7}$$

It is not difficult to see that the minimal ω satisfying these constraints corresponds to $\max_i |\gamma_i|$. Consequently, by combining the variables ω and $\vec{\gamma}$ into one variable

$$\vec{z} = \begin{pmatrix} \omega \\ \vec{\gamma} \end{pmatrix},$$

we may rewrite the minimisation in problem (6) as the following linear program with complexity $\mathcal{O}((m+1)^3)$ [16]:

$$\min_{\vec{z}} \begin{bmatrix} 1 & \vec{0}_m^T \end{bmatrix} \vec{z}, \text{ s.t. } \begin{cases} \begin{pmatrix} \vec{0}_n & \underline{G} \end{pmatrix} \vec{z} = \vec{p} - \vec{c}, \\ \begin{pmatrix} -\vec{1}_m & \underline{I}_{m\times m} \\ -\vec{1}_m & -\underline{I}_{m\times m} \end{pmatrix} \vec{z} \leq \vec{0}_{2m}. \end{cases} \tag{8}$$

The minimum $\|\vec{\gamma}\|_\infty$ of problem (6) is then stored in z_1, whereas the minimiser $\vec{\gamma} = (\gamma_1, ..., \gamma_m)$ is given by $z_2, ..., z_{m+1}$.

For later use, let us state some equivalent properties involving $\nu(\vec{p})$:

Lemma 2. *Let Z be a zonotope centred at the origin with generator matrix $\underline{G} \in \mathbb{R}^{n\times m}$, $m \geq n$. Then the following properties are equivalent:*

1. *$\underline{G}$ has rank n.*
2. *$Z^\circ \neq \emptyset$, i.e. Z has a topological interior.*
3. *$\underline{G}$ is surjective.*
4. *ν is defined for all points in $\mathbb{R}^n$. Furthermore, ν is positive definite, i.e., for any $\vec{v} \in \mathbb{R}^n$ there holds*

$$\nu(\vec{v}) = 0 \Leftrightarrow \vec{v} = \vec{0}_n. \tag{9}$$

Proof. $1 \Leftrightarrow 3$ is clear by definition. $3 \Leftrightarrow 2$ follows from the fact that Z has a topological interior if and only if Z° is locally homeomorphic to $\mathbb{R}^n$, which holds if and only if $\underline{G}$ is surjective since $m \geq n$.

Let us now show that $3 \Leftrightarrow 4$. We begin by showing that $3 \Rightarrow 4$. Therefore, assume that $\underline{G}$ is surjective, which implies that ν is defined for all points in $\mathbb{R}^n$. If $\vec{v} = \vec{0}_n$, we are searching for a $\vec{\gamma}$ such that

$$\vec{v} = \vec{0}_n = \underline{G}\vec{\gamma}. \tag{10}$$

Clearly, $\vec{\gamma} = \vec{0}_m$ is a solution and has the minimum possible length w.r.t. the ∞-norm. Therefore,

$$\nu(\vec{v}) = 0. \tag{11}$$

It, therefore, remains to show that

$$\vec{v} \neq \vec{0}_n \Rightarrow \nu(\vec{v}) \neq 0. \tag{12}$$

Again, we are looking for a $\vec{\gamma}$ such that

$$\vec{v} = \underline{G}\vec{\gamma}. \tag{13}$$

Such a $\vec{\gamma}$ exists since $\underline{G}$ is surjective, and $\vec{\gamma} \neq \vec{0}_m$ since $\underline{G}\,\vec{0}_m = \vec{0}_n \neq \vec{v}$. Thus, $\|\vec{\gamma}\|_\infty > 0$ since $\|\cdot\|_\infty$ is a norm and is, therefore, positive definite. We conclude that $\nu(\vec{v}) \neq 0$.

We now show $4 \Rightarrow 3$. Suppose, therefore, that 4 holds, but $\underline{G}$ is not surjective. Then, there is a point $\vec{p}$ that has no preimage under $\underline{G}$, and thus, for which ν is not defined, contradicting our assumption. Consequently, $\underline{G}$ needs to be surjective.

Zonotopes $Z = \langle \vec{c}, \underline{G} \rangle$ for which $\underline{G}$ has rank n are often referred to as *non-degenerate*.

3.2. Zonotope Norms

We can now define a special norm for any non-degenerate zonotope:

Theorem 1. *Let $\underline{G} \in \mathbb{R}^{n\times m}$ be a matrix of rank n. Consider the function $\nu : \mathbb{R}^n \to [0, \infty)$ defined in (6). Then, ν is a norm on $\mathbb{R}^n$.*

Proof. By Lemma 2, if $\underline{G}$ has full rank, ν is well-defined on $\mathbb{R}^n$, and positive definite. The triangle inequality $\nu(\vec{v} + \vec{w}) \leq \nu(\vec{v}) + \nu(\vec{w})$ and the positive homogeneity $\nu(a\vec{v}) = |a|\nu(\vec{v})$, for $\vec{v}, \vec{w} \in \mathbb{R}^n$ and $a \in \mathbb{R}$, follow from the fact that the constraints on the minimiser $\vec{\gamma}$ are linear and $\|\cdot\|_\infty$ is a norm.

Since this norm is handy in our context, we give it a proper definition and notation:

Definition 4. Let $Z = \langle \vec{c}, \underline{G} \rangle$ be a non-degenerate zonotope, and let ν be the function as defined in Theorem 1 for the translated zonotope $Z' = \langle \vec{0}_n, \underline{G} \rangle$. We define the induced zonotope norm

$$\|\vec{p}\|_Z := \nu(\vec{p}), \tag{14}$$

and call this norm the Z-norm.

We now discuss a few other properties of this norm:

Corollary 1. *Let $Z = \langle \vec{c}, \underline{G} \rangle$ be a non-degenerate zonotope. For $r > 0$, define the* ball $B_r^Z(\vec{c})$ *of radius r centred at $\vec{c}$ as*

$$B_r^Z(\vec{c}) := \{\vec{x} \in \mathbb{R}^n | \|\vec{x} - \vec{c}\|_Z \leq r\}. \tag{15}$$

Similarly, define the circle $C_r^Z(\vec{c}) = \partial B_r^Z(\vec{c})$ *of radius r centred at $\vec{c}$. Then,*

$$B_1^Z(\vec{c}) = Z, \quad C_1^Z(\vec{c}) = \partial Z. \tag{16}$$

Proof. This follows directly from (6).

Remark 1. *One might be tempted to think that the unit circle $C_1^Z(\vec{0}_n)$ is the image of $\partial[-1,1]^m$ under the linear map given by the matrix $\underline{G}$. However, while $\underline{G}\,(\partial[-1,1]^m) \supseteq C_1^Z(\vec{0}_n)$ always holds, the converse is not necessarily true.*

Corollary 2. *Let Z be a non-degenerate zonotope centred at the origin. Then the Z-norm coincides with the polyhedral norm of Z, i.e.,*

$$\forall \vec{p} \in \mathbb{R}^n, \ \|\vec{p}\|_Z = \mathcal{S}_Z(\vec{p}). \tag{17}$$

Proof. According to Lemma 1 and Corollary 1, the two functions have the same unit ball Z, which is symmetric around its centre since it is a zonotope. Hence, $\mathcal{S}_Z$ is symmetric around $\vec{0}_n$. From Lemma 1, it follows that $\mathcal{S}_Z$ is a norm. Since two norms that have the same unit ball are equal, we have proven the Corollary.

Remark 2. *Corollary 2 is not true when Z is degenerate. Indeed, in this case $\|\cdot\|_Z$ may not be defined on all of $\mathbb{R}^n$, unlike $\mathcal{S}_Z$. However, even in that case, $\mathcal{S}_Z$ is a norm, and can be used instead of $\|\cdot\|_Z$ to check whether a point $\vec{p} \in \mathbb{R}^n$ lies in Z.*

Corollary 3. *Let Z be a non-degenerate zonotope. Then the induced norm $\|\cdot\|_Z$ is equivalent to all p-norms in $\mathbb{R}^n$, thus $\|\cdot\|_Z$ is convex, Lipschitz-continuous on $\mathbb{R}^n$ and differentiable almost everywhere.*

Proof. All norms are convex, thus so is $\|\cdot\|_Z$. Since all norms on $\mathbb{R}^n$ are equivalent, $\|\cdot\|_Z$ is equivalent to any of the p-norms, so that in particular $\|\cdot\|_Z \leq C\|\cdot\|_2$, with C depending only on n and Z, proving that $\|\cdot\|_Z$ is Lipschitz-continuous. Lipschitz-continuity implies differentiability almost everywhere.

3.3. co-NP-Completeness of the Zonotope Containment Problem

We now turn to the more general zonotope containment problem: Given two zonotopes Z_1 and Z_2 in $\mathbb{R}^n$, the goal is to check whether $Z_1 \subseteq Z_2$. To the best knowledge of the authors, the only existing algorithms solving this problem rely either on checking containment for each vertex or facet of Z_1 (see [8]) or on an approximation algorithm (see [9, Theorem 3]) that is based on a necessary but not sufficient condition for containment to hold. The question of whether a polynomial algorithm exists for the zonotope containment problem remained unanswered in the literature.

The containment problem as defined above is a decision problem (the zonotope is either contained or not). To properly analyse the complexity of the problem, it will prove to be useful to reformulate it as an optimisation problem. To do so, note that $Z_1 \subseteq Z_2$ if and only if every point $\vec{x} \in Z_1$ also belongs to Z_2. Therefore, using the results we found in section 3.1, we can reformulate the zonotope containment in the following way:

Input: Two zonotopes $Z_1 = \langle \vec{c}_1, \underline{G}_1 \rangle$ and $Z_2 = \langle \vec{c}_2, \underline{G}_2 \rangle$ in $\mathbb{R}^n$ with m_1 and m_2 generators, respectively.

Question: Does it hold for every $\vec{p} \in Z_1$ that

$$\|\vec{p} - \vec{c}_2\|_{Z_2} \leq 1 \quad ? \tag{18}$$

We shall refer to this problem as ZC. It is in co-NP since given a point $\vec{p} \in Z_1$, we can check in polynomial time whether $\|\vec{p} - \vec{c}_2\|_{Z_2} > 1$ (i.e., whether $\vec{p} \notin Z_2$, implying that $Z_1 \not\subseteq Z_2$). To show that ZC is co-NP-hard, it suffices to show that it is co-NP-hard to estimate the value

$$d(Z_1, Z_2) := \max_{\vec{p} \in Z_1} \|\vec{p} - \vec{c}_2\|_{Z_2}, \tag{19}$$

which is an optimization problem. In the case, where Z_1 is the hypercube $\vec{c} + r[-1,1]^n$ with radius $r > 0$ centred at the same point $\vec{c}_1 = \vec{c}_2 =: \vec{c}$ as Z_2, it follows that $d(\vec{c} + r[-1,1]^n, Z_2) = 1$ if and only if the hypercube is contained in Z_2 and touches its boundary. In this case, $r > 0$ gives the length of the largest hypercube contained within Z_2, meaning that r is the inradius of Z_2 w.r.t. the ∞-norm, so that the inradius $r_\infty(Z)$ of a zonotope $Z = \langle \vec{c}, \underline{G} \rangle$ w.r.t. the ∞-norm is

$$r_\infty(Z) = \frac{1}{d(\vec{c} + [-1,-1]^n, Z)}. \tag{20}$$

Thus, we turn towards the following problem, which we refer to as INRAD_∞:

Input: A zonotope $Z \subset \mathbb{R}^n$, a positive number $k \in \mathbb{R}^+$.

Question: Does $r_\infty(Z) \leq k$ hold?

If INRAD_∞ is NP-hard, it implies that ZC is co-NP-hard if Z_1 is a hypercube, since $d(\vec{c} + [-1,-1]^n, Z) \geq k$ if and only if $r_\infty(Z) \leq k$. Since hypercubes are special types of zonotopes, we would have shown that the general zonotope containment problem ZC is co-NP-hard. To prove that INRAD_∞ is NP-hard, we first need some technical lemmas:

Lemma 3. *Let $\vec{b} \in \mathbb{R}^n$, $H = \left\{ \vec{x} \in \mathbb{R}^n \middle| \vec{b}^T \vec{x} \leq 1 \right\}$ and suppose that H contains the origin. Then the largest hypercube centred at the origin $C = r[-1,1]^n$, $r > 0$ that is fully contained within H has radius*

$$r = \frac{1}{\|\vec{b}\|_1}. \tag{21}$$

Proof. The largest hypercube contained in H touches the boundary $\vec{b}^T \vec{x} = 1$ of H, thus at least one vertex $r\vec{v}$ of C touches the boundary of H, where $\vec{v} \in \{-1,1\}^n$. This translates to the condition

$$r \vec{b}^T \vec{v} = 1. \tag{22}$$

Since by assumption $r \geq 0$ is the minimal radius s.t. (22) holds, we can check (22) for each $\vec{v} \in \{-1,1\}^n$ and choose the one for which r has the smallest value, which can be formulated as

$$r = \min_{\vec{v} \in \{-1,1\}^n} \frac{1}{|\vec{b}^T \vec{v}|} = \frac{1}{\max_{\vec{v} \in \{-1,1\}^n} |\vec{b}^T \vec{v}|}. \tag{23}$$

The Lemma then follows from

$$\max_{\vec{v} \in \{-1,1\}^n} |\vec{b}^T \vec{v}| = \|\vec{b}\|_1. \tag{24}$$

Lemma 4. *Let the polytope $P \subset \mathbb{R}^n$ be bounded, non-degenerate (i.e., with non-empty topological interior), centrally symmetric around the origin and let P^Δ be its dual polytope. The circumradius $R_1(P)$ of P w.r.t. the 1-norm and the inradius $r_\infty(P^\Delta)$ of P^Δ w.r.t. the ∞-norm are reciprocal:*

$$R_1(P) = \frac{1}{r_\infty(P^\Delta)}. \tag{25}$$

Proof. Let $P = \mathrm{conv}(\vec{v}_1, \dots, \vec{v}_N)$ be a V-representation of P. Since the 1-norm is a convex function, its maximum over a polytope is attained at one of its vertices. Since P contains the origin, the dual P^Δ of P is given as [11, Theorem 2.11. (vi), p.62]

$$P^\Delta = \left\{ \vec{x} \in \mathbb{R}^n \middle| \vec{v}_i^T \vec{x} \leq 1, i = 1, \dots, N \right\}. \tag{26}$$

P^Δ is also centrally symmetric, so the inradius of P^Δ w.r.t. the ∞-norm is the radius $r_\infty(P^\Delta)$ of the largest hypercube $r_\infty(P^\Delta)[-1,1]^n$ that is entirely contained in P^Δ. For each halfspace $H_i = \{\vec{x} \in \mathbb{R}^n | \vec{v}_i^T \vec{x} \leq 1\}$, we know from Lemma 3 that this radius can be at most

$$r_i := \frac{1}{\|\vec{v}_i\|_1}. \tag{27}$$

Therefore, $r_\infty(P^\Delta) = \min_i r_i$, which means that

$$r_\infty(P^\Delta) = \min_i \frac{1}{\|\vec{v}_i\|_1} = \frac{1}{\max_i \|\vec{v}_i\|_1} = \frac{1}{R_1(P)}. \tag{28}$$

We now turn towards the main construction required for proving that computing $r_\infty(Z)$ is NP-hard, closely following [17, Theorem 2.11., p.229]. The main idea is to construct a zonotope that has the same inradius w.r.t. the ∞-norm as the dual polytope of a given parallelotope. Since computing the circumradius of a parallelotope is NP-hard (see [18, Theorem 3.5, p.20]), using Lemma 4 implies that $r_\infty(Z)$ is NP-hard to compute as well.

Let $\vec{z}_1, \dots, \vec{z}_n$ be linearly independent vectors in $\mathbb{R}^n$ spanning a parallelotope Π. With $\vec{v}_i = \vec{z}_i$ and $\vec{v}_{n+i} = -\vec{z}_i$ for $i = 1, \dots, n$, let $\underline{V}$ be the matrix whose rows are the vectors $\vec{v}_j$ so that we can write Π as

$$\Pi = \sum_{i=1}^{n} [-1,1]\, \vec{z}_i = \left\{ \vec{x} \in \mathbb{R}^n \middle| \underline{V} \vec{x} \leq \vec{1} \right\}. \tag{29}$$

Since Π contains the origin, the polar dual Π^Δ of Π is given by $\mathrm{conv}(\vec{v}_1, \dots, \vec{v}_{2n})$ [11, Theorem 2.11. (vii), p.62]. We also make use of the fact that the circumradius $R_\infty(\Pi)$ of Π w.r.t. the ∞-norm can be computed in polynomial time (see [19, Proposition

2.2, p.16]). By the equivalence of the norms $\|\cdot\|_1$ and $\|\cdot\|_\infty$, we have that

$$R_1(\Pi) = \max_i \|\vec{v}_i\|_1 \leq \max_i n\|\vec{v}_i\|_\infty = nR_\infty(\Pi). \tag{30}$$

We then define

$$\alpha := \frac{1}{2n^2}\frac{1}{R_\infty(\Pi)}. \tag{31}$$

By using (30) on (31) we deduce

$$\alpha \leq \frac{1}{2n}\frac{1}{R_1(\Pi)} = \frac{r_\infty(\Pi^\Delta)}{2n}. \tag{32}$$

We now have the necessary tools to construct the zonotope used to prove the NP-hardness of INRAD_∞:

$$Z^* = \sum_{i=1}^{2n}[0,1](\vec{v}_i + \alpha\vec{e}_{n+1}). \tag{33}$$

This is a zonotope (see also [20, Equation (9.2), p. 167]) with generators $\vec{g}_i := \frac{1}{2}(\vec{v}_i + \alpha\vec{e}_{n+1})$ and centre $\vec{c} := n\alpha\vec{e}_{n+1}$. Furthermore, as proven in [17, p.230], it contains an embedding of Π^Δ in $\mathbb{R}^{n+1}$:

$$\Pi^\Delta \times [\alpha, (2n-1)\alpha] \subset Z^*. \tag{34}$$

Lemma 5. *The inclusion (34) is tight in the sense that $\Pi^\Delta \times [\alpha, (2n-1)\alpha]$ touches the boundary of Z^*:*

$$\Pi^\Delta \times \{\alpha\} = Z^* \cap (\mathbb{R}^n \times \{\alpha\}). \tag{35}$$

More generally, for $\beta \in [0, \alpha]$, it holds that

$$\frac{\beta}{\alpha}\Pi^\Delta \times \{\beta\} = Z^* \cap (\mathbb{R}^n \times \{\beta\}), \tag{36}$$

where $\frac{\beta}{\alpha}\Pi^\Delta$ is to be understood as the set one obtains by scaling each element of Π^Δ by $\frac{\beta}{\alpha}$.

Proof. Since (35) follows from (36) for $\beta = \alpha$, it suffices to show (36). Let $\beta \in [0, \alpha]$.

Claim: $\frac{\beta}{\alpha}\Pi^\Delta \times \{\beta\} \subseteq Z^* \cap (\mathbb{R}^n \times \{\beta\})$

Let $\vec{x} = \sum_{i=1}^{2n} \lambda_i \frac{\beta}{\alpha}\vec{v}_i + \beta\vec{e}_{n+1} \in \frac{\beta}{\alpha}\Pi^\Delta \times \{\beta\}$, with $\sum_{i=1}^{2n}\lambda_i = 1$ and $\lambda_i \geq 0$ for $i = 1, ..., 2n$. For $\mu_i := \frac{\beta}{\alpha}\lambda_i$ we have that $\mu_i \in [0,1]$ for $i = 1, ..., 2n$, and since $1 = \sum_{i=1}^{2n}\lambda_i$ we may write

$$\vec{x} = \sum_{i=1}^{2n}\lambda_i\frac{\beta}{\alpha}\vec{v}_i + \sum_{i=1}^{2n}\lambda_i\beta\vec{e}_{n+1} = \sum_{i=1}^{2n}\mu_i(\vec{v}_i + \alpha\vec{e}_{n+1}), \tag{37}$$

which shows that $\vec{x} \in Z^*$. Since $\vec{x} \in \mathbb{R}^n \times \{\beta\}$, we have proven that $\frac{\beta}{\alpha}\Pi^\Delta \times \{\beta\} \subseteq Z^* \cap (\mathbb{R}^n \times \{\beta\})$.

Claim: $\frac{\beta}{\alpha}\Pi^\Delta \times \{\beta\} \supseteq Z^* \cap (\mathbb{R}^n \times \{\beta\})$

Let $\vec{x} = \sum_{i=1}^{2n}\mu_i(\vec{v}_i + \alpha\vec{e}_{n+1}) \in Z^*$ for $\mu_i \in [0,1]$, $i = 1, ..., 2n$ be such that $\vec{x}^T\vec{e}_{n+1} = \beta$, which implies that $\sum_{i=1}^{2n}\mu_i\alpha = \beta$. With $\lambda_i = \frac{\alpha}{\beta}\mu_i$, we have $\sum_{i=1}^{2n}\lambda_i = 1$ and $\vec{x} = \sum_{i=1}^{2n}\lambda_i\vec{v}_i + \beta\vec{e}_{n+1} \in \frac{\beta}{\alpha}\Pi^\Delta \times \{\beta\}$, showing that $\frac{\beta}{\alpha}\Pi^\Delta \times \{\beta\} \supseteq Z^* \cap (\mathbb{R}^n \times \{\beta\})$.

Let C be the maximal hypercube centred around $\vec{c}$ that is entirely contained within Z^*. By definition, C has radius $r_\infty(Z^*)$. For our next results, we refer to $|\vec{e}_{n+1}^T(\vec{p}_1 - \vec{p}_2)|$ for two points $\vec{p}_1, \vec{p}_2 \in \mathbb{R}^{n+1}$ as the *height distance* of these two points.

Lemma 6. *There is a vertex $\vec{q}$ of C that touches the boundary of Z^*, and for which $\vec{q}^T\vec{e}_{n+1} \in [0, \alpha]$.*

Proof. A point $\vec{x} = (\vec{s}, t) \in Z^*$ must lie in exactly one of the following zones of Z^*, see also Figure 1:

I) The (topological) interior of $\Pi^\Delta \times [\alpha, (2n-1)\alpha]$,
II) $\|\vec{s}\|_\infty > r_\infty(\Pi^\Delta)$,
III) $\|\vec{s}\|_\infty \leq r_\infty(\Pi^\Delta)$ and $t \in [(2n-1)\alpha, 2n\alpha]$,
IV) $\|\vec{s}\|_\infty \leq r_\infty(\Pi^\Delta)$ and $t \in [0, \alpha]$.

By definition (31) of α, it follows that

$$(2n-2)\alpha = \frac{2n-2}{2n}r_\infty(\Pi^\Delta) \leq r_\infty(\Pi^\Delta), \tag{38}$$

i.e., the height of $\Pi^\Delta \times [\alpha, (2n-1)\alpha]$ w.r.t. $\vec{e}_{n+1}$ is smaller than $r_\infty(\Pi^\Delta)$. Since C is the largest hypercube in Z^* centred at $\vec{c}$, at least one of its vertices

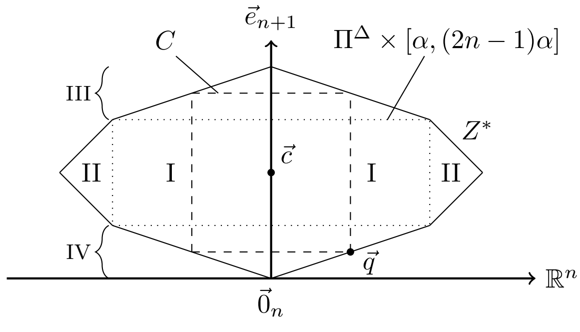

Figure 1: A two-dimensional projection of Z^*. The dotted line represents the embedded $\Pi^\Delta \times [\alpha, (2n-1)\alpha]$, the dashed one the hypercube C, and the numbers I, II, III, and IV refer to the zones described in the proof of Lemma 6. If α is small enough such that the height of $\Pi^\Delta \times [\alpha, (2n-1)\alpha]$ is smaller than its width, C is guaranteed to touch ∂Z^* at a point $\vec{q}$ of height smaller than α.

must touch the boundary of Z^* and, thus, must lie outside the interior of $\Pi^\Delta \times [\alpha, (2n-1)\alpha]$, meaning that it must have a height distance from $\vec{c}$ larger than the height of $\Pi^\Delta \times [\alpha, (2n-1)\alpha]$. Since C is symmetric, this means that each of its vertices must have the same height distance to the centre, implying that every vertex of C must lie outside the interior of $\Pi^\Delta \times [\alpha, (2n-1)\alpha]$, i.e., outside zone I. Then, since (34) is tight in the sense discussed in Lemma 5, we conclude that

$$r_\infty(Z^*) \leq \min\{r_\infty(\Pi^\Delta), (2n-2)\alpha\}, \tag{39}$$

which, due to (38) and (39), means that

$$r_\infty(Z^*) \leq r_\infty(\Pi^\Delta). \tag{40}$$

Therefore, if there was a vertex $\vec{q} = (\vec{s}, t)$ of C in zone II, i.e., such that $\|\vec{s}\|_\infty > r_\infty(\Pi^\Delta)$, it would follow by the definition of the ∞-norm that $\|\vec{q}\|_\infty > r_\infty(\Pi^\Delta)$, contradicting the fact that any vertex of C must have norm $r_\infty(Z^*) \leq r_\infty(\Pi^\Delta)$. Consequently, $\|\vec{s}\|_\infty \leq r_\infty(\Pi^\Delta)$ must hold. Now, assume $\vec{q} = (\vec{s}, t)$ is a vertex of C that touches the boundary of Z^*. If $\vec{q}$ is in zone III, i.e., $t \in [(2n-1)\alpha, 2n\alpha]$, then mirroring $\vec{q}$ w.r.t. the centre $\vec{c}$ is again a vertex $\vec{q}\,'$ of C due to the symmetry of hypercubes, and now, $\vec{q}\,'^T \vec{e}_{n+1} \in [0, \alpha]$. The point $\vec{q}\,'$ also touches the boundary of Z^* because of the symmetry of Z^*. Thus, in any case, we can find a vertex of C that touches the boundary of Z^* in zone IV, i.e., that has height distance from the origin smaller than α.

Theorem 2. *$INRAD_\infty$ is NP-hard.*

Proof. Let Z^* and C be as described above, and $\vec{q}$ a vertex as described in Lemma 6. Since it is a vertex of C, it can be expressed as

$$\vec{q} = \vec{c} + r_\infty(Z^*) \begin{pmatrix} \vec{\sigma}^T & -1 \end{pmatrix}^T, \tag{41}$$

where $\vec{\sigma}$ is some point $\vec{\sigma} \in \{-1, 1\}^n$. Therefore, the height of $\vec{q}$ is $\vec{q}^T \vec{e}_{n+1} = \vec{c}^T \vec{e}_{n+1} - r_\infty(Z^*) = n\alpha - r_\infty(Z^*)$. Now, for $\beta = n\alpha - r_\infty(Z^*)$ consider the point $\vec{p} \in \frac{\beta}{\alpha}\Pi^\Delta$ given as $\vec{q}^T = \begin{pmatrix} \vec{p}^T & q_{n+1} \end{pmatrix}$ (i.e., $\vec{p}$ corresponds to the first n coordinates of $\vec{q}$), which exists thanks to (36). It has to lie on the boundary of $\frac{\beta}{\alpha}\Pi^\Delta$ since $\vec{q}$ is on the boundary of Z^*, and since the boundary of $Z^* \cap (\mathbb{R}^n \times \{\beta\})$ and $\frac{\beta}{\alpha}\Pi^\Delta \times \{\beta\}$ coincide because of (36). From (41), it also follows that $\vec{p}$ is the vertex of a hypercube centred at the origin, corresponding to the scaled lower face of the hypercube C, which is, therefore, contained entirely within $\frac{\beta}{\alpha}\Pi^\Delta$ since

$$C \cap (\mathbb{R}^n \times \{\beta\}) \subseteq Z^* \cap (\mathbb{R}^n \times \{\beta\}) = \frac{\beta}{\alpha}\Pi^\Delta \times \{\beta\}. \tag{42}$$

Therefore, $\vec{p}$ is a vertex of a hypercube entirely contained within $\frac{\beta}{\alpha}\Pi^\Delta$, that touches the boundary $\frac{\beta}{\alpha}\partial\Pi^\Delta$. Consequently, the hypercube in question is the largest hypercube centred around the origin entirely contained within $\frac{\beta}{\alpha}\Pi^\Delta$, meaning that

$$\begin{aligned} &\frac{\beta}{\alpha} r_\infty(\Pi^\Delta) = \|\vec{p}\|_\infty \\ \Leftrightarrow \quad &r_\infty(\Pi^\Delta) = \frac{\alpha r_\infty(Z^*)}{n\alpha - r_\infty(Z^*)}, \end{aligned} \tag{43}$$

where we used the fact that $\|\vec{\sigma}\|_\infty = 1$. We finally conclude that using Lemma 4, there holds

$$R_1(\Pi) = \frac{n\alpha - r_\infty(Z^*)}{\alpha r_\infty(Z^*)}. \tag{44}$$

The function

$$f(x) = \frac{n\alpha - x}{\alpha x} \tag{45}$$

is smooth and strictly decreasing for $x > 0$ and, thus, is bijective onto its image for $x > 0$. Therefore, we infer using (44) that to check whether $R_1(\Pi) \geq k$ for some $k > 0$ is equivalent to check whether $r_\infty(Z^*) \leq \frac{n\alpha}{\alpha k - 1}$. Since checking whether $R_1(\Pi) \geq k$ is NP-hard by [18, Theorem 3.5, p.20], it follows that INRAD_∞ is also NP-hard.

Corollary 4. *ZC is co-NP-complete.*

As previously mentioned, [7] showed that the containment problem is actually solvable in polynomial time w.r.t. the number of facets f_1 and f_2 of Z_1 and Z_2. However, the true computational load of the problem is not necessarily reflected by this fact: Since the number of facets of a zonotope can grow exponentially w.r.t. the number of generators, it is usually costly to store the facets of a zonotope as opposed to storing the generators. In that regard, Corollary 4 shows that the high computational cost that arises in practice for the zonotope containment problem can essentially not be efficiently reduced by considering a simpler representation for zonotopes.

4. Algorithms for the Zonotope Containment Problem

Let Z_1 and Z_2 be zonotopes in $\mathbb{R}^n$ with m_1 and m_2 generators, respectively. As we have seen, the zonotope containment problem is in general co-NP-complete. However, if one of the quantities m_1, m_2 or n remains fixed and we consider scalability w.r.t. the other two, one can indeed find algorithms that run in polynomial time.

Fixed m_1

If m_1 is fixed, Z_1 has a fixed upper bound of 2^{m_1} vertices. For each of these, we can compute its distance to $\vec{c}_2$ w.r.t. the Z_2-norm, which can be done in polynomial time (see Algorithm 1). $Z_1 \subseteq Z_2$ if and only if there is no vertex with a norm larger than one. We call this procedure `venum`.

Algorithm 1 Vertex enumeration (`venum`)

Input: Zonotopes $Z_1 = \langle \vec{c}_1, \underline{G}_1 \rangle$ and $Z_2 = \langle \vec{c}_2, \underline{G}_2 \rangle$.
Output: True if $Z_1 \subseteq Z_2$, False otherwise.

```
V ← vertices(Z1)
for v ∈ V do
    if ‖v − c2‖_Z2 > 1 then
        return False
    end if
end for
return True
```

Fixed m_2

Z_2 has at most $2\binom{m_2}{n-1}$ facets. Since the polyhedral norm can be computed in polynomial time w.r.t. the number of facets of Z_2, using the algorithm in [13, p. 269], we can solve ZC in polynomial time w.r.t. the number of facets of Z_2 as demonstrated in Algorithm 2. We refer to this algorithm as `polymax`. For fixed m_2, the number of facets is bounded for varying n, since for $n > m_2$, the zonotope becomes degenerate, which implies that `polymax` has polynomial runtime. Note that the bulk of the computation time for `polymax` comes from the computation of the halfspace representation of Z_2, not the actual norm-maximisation.

Fixed n

If n is fixed, we use the same method as for the case where m_2 is fixed, except that the number of facets can be bounded as

$$2\binom{m_2}{n-1} \leq 2\left(\frac{e\,m_2}{n-1}\right)^{n-1}, \tag{46}$$

where e is Euler's number. Therefore, the number of facets increases at most as m_2^{n-1}, which is polynomial in m_2 since n is fixed.

Algorithm 2 Maximal polyhedral norm (`polymax`), see also [13, p.269]

Input: Zonotopes $Z_1 = \langle \vec{c}_1, \underline{G}_1 \rangle$ and $Z_2 = \langle \vec{c}_2, \underline{G}_2 \rangle$.
Output: True if $Z_1 \subseteq Z_2$, False otherwise.

```
(H, h) ← halfspace(Z2)
η1, ..., ηs row vectors of H
g1, ..., g_m1 column vectors of G1
for i = 1, ..., s do
    for j = 1, ..., m1 do
        x_j ← { 1,  if (v_i^T g_j)/h_i ≥ 0
              { −1, otherwise
    end for
    if ‖x − c2‖_Z2 > 1 then
        return False
    end if
end for
return True
```

Optimisation Algorithm

While `venum` and `polymax` work well and are typically fast if either m_1 or the number of facets of Z_2 is low, these algorithms are each tailored for a specific case, i.e., a certain quantity that remains fixed. There is another method of solving the zonotope containment problem that does not require any assumptions to work well, even though, it is comparably slow for low n, m_1, m_2.

Indeed, we can solve the containment problem by computing $d(Z_1, Z_2)$ as defined in (19). Since the maximum of a convex function over a polytope is reached at one of its vertices, and since $\|\cdot\|_{Z_2}$ is convex by Corollary 3, we can compute $d(Z_1, Z_2)$ by maximizing the following function:

$$f(\vec{x}) = \|\underline{G}_2\vec{x} + \vec{c}_1 - \vec{c}_2\|_{Z_2}, \quad \text{s.t. } \vec{x} \in \{-1, 1\}^{m_1}. \tag{47}$$

In this study, maximising $f(\vec{p})$ was done with the MATLAB function `surrogateopt`, which evaluates the function on random points to construct a surrogate function that is then used to find a maximum. This solver is guaranteed to converge to a global maximum [21], however there is no clear stopping criterion other than the maximum number of function evaluations that have been performed, or checking whether the algorithm has found a point $\vec{p}$ such that $f(\vec{p}) > 1$. As a consequence, this algorithm is efficient in particular for showing that $Z_1 \not\subseteq Z_2$, since the algorithm is guaranteed to terminate in that case. In this study, we set the maximum number N of function evaluations to 500. We refer to this algorithm as `opt`.

5. Experimental Results

5.1. Runtime Comparison

Experimental results for the runtime of each method described in section 4 are shown in Figures 2-4, along with the runtime of the algorithm from [9, Theorem 1.], which we denote by `st`, and that returns an approximative result. In Figure 5, the performance of `opt` is compared to that of `st` on cases where $m_1 = m_2 = 2n$. All computations have been performed in MATLAB on an Intel Core i7-8650U CPU @1.9GHz with 24GB memory.

For any choice of m_1, m_2, and n, each algorithm was tested on 100 zonotope pairs $Z_1 = \langle \vec{c}_1, \underline{G}_1 \rangle$, $Z_2 = \langle \vec{c}_2, \underline{G}_2 \rangle$ that were generated randomly using two different methods, forming two groups of 50 zonotope pairs each. For both groups, the entries of the generator matrices $\underline{G}_1$ and $\underline{G}_2$ were sampled uniformly in $[-1, 1]$. For pairs of group 1, $\underline{G}_1$ and $\underline{G}_2$ were then scaled in such a way that $R_\infty(Z_1) = 0.1$ and $R_\infty(Z_2) = 1$, with R_∞ defined as in section 3.3. The centre $\vec{c}_2$ of Z_2 was chosen to be the origin, whereas the centre $\vec{c}_1$ of Z_1 was sampled uniformly in $[-0.1, 0.1]^n$. In group 2, the centres of both Z_1 and Z_2 were chosen to be the origin, and $\underline{G}_1$ and $\underline{G}_2$ were scaled such that $R_\infty(Z_1) = R_\infty(Z_2) = 1$. Note that pairs in group 1 are more likely to satisfy the containment problem, i.e., $Z_1 \subseteq Z_2$ is more likely to happen, while for pairs in group 2, $Z_1 \not\subseteq Z_2$ is more likely. Specifically, for all zonotope pairs that we tested for Figures 2-4, all pairs in group 2 satisfied $Z_1 \not\subseteq Z_2$, and all pairs in group 1 with $m_2 > 8 + n$ satisfied $Z_1 \subseteq Z_2$. Furthermore, note that both `st` and `opt` gave the correct solution to the containment problem for all these pairs.

In each of the Figures 2-5, the graph on the top shows the worst-case runtime for group 1, whereas the graph on the bottom shows the worst-case runtime for group 2. Note that the scaling of the y-axis differs for both groups. In Figure 2, one can see that `opt` is as fast as `venum` for group 1, but faster for group 2. For low dimensions, `opt` is typically slower than `polymax`, but this changes for higher dimensions, in particular for pairs in group 2, as can be seen in Figure 4.

If $m_1 = m_2 = 2n$, both the number of vertices and the number of facets of Z_1 and Z_2 can grow exponentially w.r.t. n. Therefore, this setting can be used to compare the overall scalability of all algorithms. For example, `polymax` completely failed to produce results for $n > 10$ on our machine, because the algorithm would have required at least 10-fold the amount of memory we had available, i.e., about 800 GB. `venum` on the other hand does not have such a high memory cost, yet as a trade-off, has a long computation time, e.g., already 120 s for $n = 6$ as opposed to less than 20 s for `opt` in the worst case. Finally, `st` performs better than `opt` for pairs in group 1, but performs worse for pairs in group 2.

We conclude that, for large n, m_1, and m_2, `st` can easily confirm whether $Z_1 \subseteq Z_2$, whereas `opt`

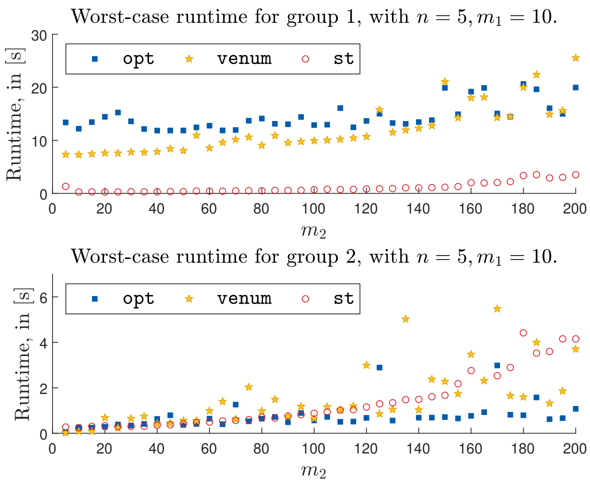

Figure 2: While all three algorithms increase for larger m_2, the computation time remains linear in m_2 if m_1 is fixed.

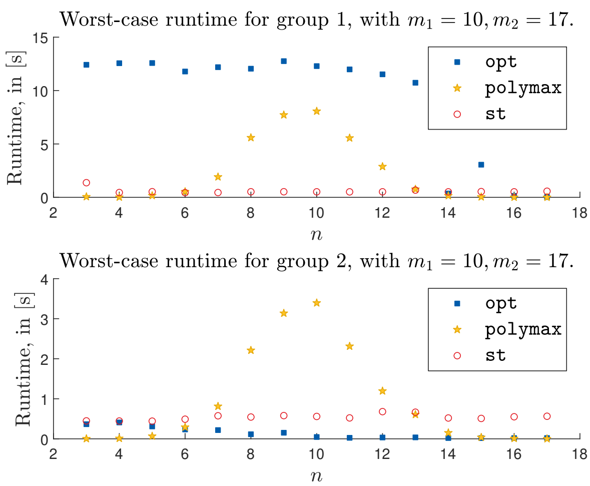

Figure 3: If m_2 is fixed, one can see that after a "bump" for `polymax`, the runtime quickly falls off as expected. The height of this bump increases exponentially w.r.t. m_2.

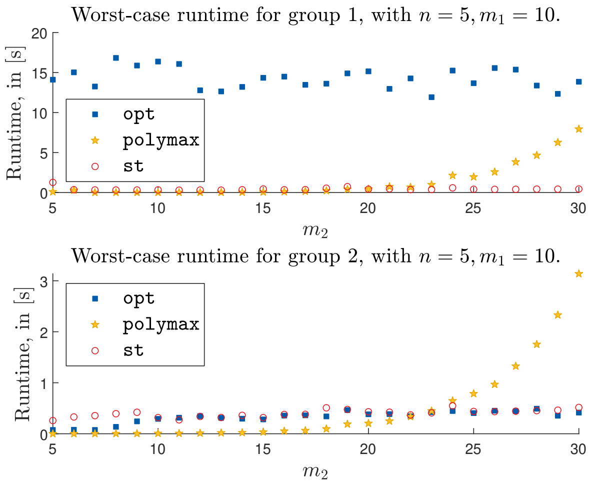

Figure 4: When n is fixed, the runtime of `polymax` is polynomial of order $\mathcal{O}(m_2^{n-1})$ according to (46). A comparison to `opt` shows that `polymax` initially performs much better but will eventually perform worse for large m_2.

is better suited to show $Z_1 \not\subseteq Z_2$. Thus, a general strategy to solve the containment problem would be to run both algorithms in parallel. Since `opt` is guaranteed to converge to the correct solution, given enough iterations, if both methods fail one can run `opt` again with a larger maximum number of function evaluations, until it yields a satisfactory result.

5.2. Accuracy of the Optimisation Approach

To conclude this section, we investigate the accuracy of `opt` compared to that of `st`. As mentioned above, both the accuracy and runtime of `opt` depend on the maximal number N of function evaluations that the solver `surrogateopt` performs. Since N is a constant that is set a-priori, the runtime of `opt` scales polynomially w.r.t. n, m_1, and m_2, since the function to be evaluated is $\|\cdot\|_{Z_2}$, which can be computed in polynomial time. The algorithm `st` can also be evaluated in polynomial time, but does not have a parameter that one can adapt to increase the accuracy of the result. The precision of `st` is therefore fixed, in contrast to `opt`, which can achieve an arbitrary precision, at the cost of a longer runtime. Using the loss function defined in [9, Section IV.A.], we can compare the precision of `opt` for $N = 500$ to that of `st`. The loss is defined as

$$\text{loss} = \frac{|\lambda_{\text{exact}} - \lambda_{\text{method}}|}{\lambda_{\text{exact}}}, \tag{48}$$

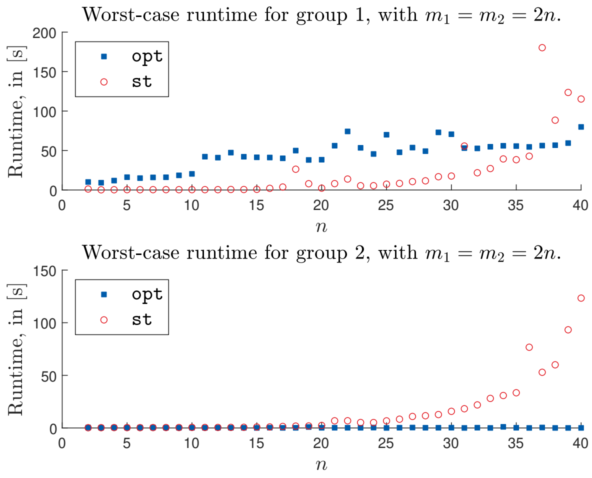

Figure 5: When $m_1 = m_2 = 2n$, comparing `opt` to `st` shows that `st` performs better when $Z_1 \subseteq Z_2$, whereas `opt` performs better when $Z_1 \not\subseteq Z_2$.

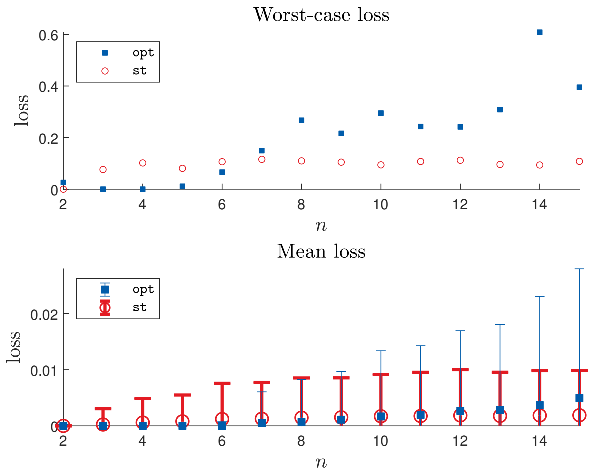

Figure 6: The graph on the top shows the worst-case loss of `st` and `opt`, whereas the graph on the bottom shows the mean loss, as well as the standard deviation over all 10000 zonotope pairs for the case $N = 500$. While `opt` is less accurate for higher dimensions, the error is of the same order of magnitude as for `st` on average.

where λ_{exact} is the maximum scaling factor such that

$$\lambda_{\text{exact}} Z_1 \subseteq Z_2, \tag{49}$$

which can be determined using either Algorithm 1 or Algorithm 2, and λ_{method} is the maximum scaling factor for which the specified method outputs that $\lambda_{\text{method}} Z_1 \subseteq Z_2$ holds. This loss was computed for 10000 zonotope pairs in dimensions $n = 2, \ldots, 15$. For each zonotope, the number of generators was uniformly sampled between n and $n + 5$, and the entries of the generator matrix were uniformly sampled in $[-1, 1]$, similarly to [9]. The results are shown in Figure 6. One can observe that the overall precision is similar to the one of `st`, but deteriorates for higher dimensions. This can be counteracted by increasing the number of function evaluations N, for example by making N depend on n, m_1, or m_2.

6. Conclusions

We have presented a fast and exact algorithm to decide whether a point belongs to a zonotope. By extrapolating this idea, we have shown that any non-degenerate zonotope induces a norm on $\mathbb{R}^n$, which can be used to describe points on the boundary of the zonotope, thereby providing information about its topology. This norm was then used for an optimisation problem that could be demonstrated to be co-NP-complete and that is equivalent to the zonotope containment problem, thus, answering the question in [9] of whether the zonotope containment problem is co-NP-complete. Since similar problems (see [22]) are even APX-hard, it would be interesting to further investigate whether the zonotope containment problem can be approximated in polynomial time, or whether such an approximation can only have limited accuracy.

We have also provided several algorithms to solve the zonotope containment problem, even though these eventually scale exponentially in general. Two of them were tailored for specific situations to prove that the problem can be solved in polynomial time if certain conditions are met, whereas the last one has proven to be efficient in higher dimensions for disproving containment, managing to solve cases that were infeasible in practice until now. Since this algorithm is based on non-linear global optimisation, it would also be of interest to compare different solvers, e.g., that are specialised for non-smooth and Lipschitz-continuous functions.

Acknowledgements

We gratefully acknowledge the financial support from the project justITSELF funded by the European Research Council (ERC) under grant agreement No 817629.

References

[1] M. Althoff, G. Frehse, Combining zonotopes and support functions for efficient reachability analysis of linear systems, in: Proc. of the 55th IEEE Conference on Decision and Control, 2016, pp. 7439–7446. doi:https://doi.org/10.1109/CDC.2016.7799418.

[2] T. Alamo, J. M. Bravo, E. F. Camacho, Guaranteed state estimation by zonotopes, Automatica 41 (6) (2005) 1035–1043. doi:https://doi.org/10.1016/j.automatica.2004.12.008.

[3] Y. Wang, V. Puig, G. Cembrano, Robust fault estimation based on zonotopic Kalman filter for discrete-time descriptor systems, International Journal of Robust and Nonlinear Control 28 (16) (2018) 5071–5086. doi:https://doi.org/10.1002/rnc.4298.

[4] J. M. Bravo, T. Alamo, E. F. Camacho, Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets, Automatica 42 (10) (2006) 1745–1751. doi:https://doi.org/10.1016/j.automatica.2006.05.003.

[5] B. Schürmann, M. Althoff, Guaranteeing constraints of disturbed nonlinear systems using set-based optimal control in generator space, in: Proc. of the 20th World Congress of the International Federation of Automatic Control, 2017, pp. 11515–11522. doi:https://doi.org/10.1016/j.ifacol.2017.08.1617.

[6] H. Roehm, J. Oehlerking, M. Woehrle, M. Althoff, Reachset conformance testing of hybrid automata, in: Proc. of Hybrid Systems: Computation and Control, 2016, pp. 277–286. doi:https://doi.org/10.1145/2883817.2883828.

[7] P. Gritzmann, V. Klee, On the complexity of some basic problems in computational convexity: I. containment problems, Discrete Mathematics 136 (1994) 129 – 174. doi:https://doi.org/10.1016/0012-365X(94)00111-U.

[8] K. Kellner, T. Theobald, C. Trabandt, Containment problems for polytopes and spectrahedra, SIAM Journal on Optimization 23 (2) (2013) 1000–1020. doi:https://doi.org/10.1137/120874898.

[9] S. Sadraddini, R. Tedrake, Linear encodings for polytope containment problems, in: IEEE 58th Conference on Decision and Control, 2019, pp. 4367–4372. doi:https://doi.org/10.1109/CDC40024.2019.9029363.

[10] M. Althoff, An introduction to CORA 2015, in: Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems, Vol. 34, 2015, pp. 120 – 151. doi:https://doi.org/10.29007/zbkv.

[11] G. Ziegler, Lectures on Polytopes, 1st Edition, Graduate Texts in Mathematics, Springer New York, 2012, updated Seventh Printing. doi:https://doi.org/10.1007/978-1-4613-8431-1.

[12] M. Althoff, O. Stursberg, M. Buss, Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes, Nonlinear Analysis: Hybrid Systems 4 (2010) 233–249. doi:https://doi.org/10.1016/j.nahs.2009.03.009.

[13] A. E. Baburin, A. V. Pyatkin, Polynomial algorithms for solving the vector sum problem, Journal of Applied and Industrial Mathematics 1 (3) (2007) 268–272. doi:https://doi.org/10.1134/S1990478907030027.

[14] M. Deza, M. D. Sikirić, Voronoi polytopes for polyhedral norms on lattices, Discrete Applied Mathematics 197 (2015) 42 – 52. doi:https://doi.org/10.1016/j.dam.2014.09.007.

[15] R. K. Ahuja, Minimax linear programming problem, Operations Research Letters 4 (3) (1985) 131 – 134. doi:https://doi.org/10.1016/0167-6377(85)

90017-3.
[16] P. M. Vaidya, An algorithm for linear programming which requires $\mathcal{O}(((m+n)n^2+(m+n)^{1.5}n)L)$ arithmetic operations, in: Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, 1987, p. 29–38.
[17] T. Burger, P. Gritzmann, Finding optimal shadows of polytopes, Discrete & Computational Geometry 24 (2000) 219–240. doi:https://doi.org/10.1007/s004540010029.
[18] H. Bodlaender, P. Gritzmann, V. Klee, J. Van Leeuwen, Computational complexity of norm-maximization, Combinatorica 10 (1990) 203–225. doi:https://doi.org/10.1007/BF02123011.
[19] M. Althoff, Reachability analysis and its application to the safety assessment of autonomous cars, Ph.D. thesis, Technical University of Munich (2010).
[20] M. Beck, S. Robins, Computing the Continuous Discretely: Integer-Point Enumeration in Polyhedra, Springer New York, 2015. doi:https://doi.org/10.1007/978-1-4939-2969-6.
[21] H. M. Gutmann, A radial basis function method for global optimization, Journal of Global Optimization 19 (2001) 201–227. doi:https://doi.org/10.1023/A:1011255519438.
[22] V. V. Shenmaier, Complexity and approximation of finding the longest vector sum, Computational Mathematics and Mathematical Physics 58 (6) (2018) 850–857. doi:https://doi.org/10.1134/S0965542518060131.