Minimum Convex Container of Two Convex Polytopes under Translations*

Hee-Kap Ahn† Sang Won Bae‡ Otfried Cheong§ Dongwoo Park† Chan-Su Shin¶

Abstract

Given two convex d-polytopes P and Q in $\mathbb{R}^d$ for $d \geq 3$, we study the problem of bundling P and Q in a smallest convex container. More precisely, our problem asks to find a minimum convex set containing P and Q that are in contact under translations. For dimension $d = 3$, we present the first exact algorithm that runs in $O(n^3)$ time, where n denotes the number of vertices of P and Q. Our approach easily extends to any higher dimension $d > 3$, resulting in the first exact algorithm.

1 Introduction

Given two convex d-polytopes P and Q in a d-dimensional space for some constant $d \geq 3$, we study the problem of *bundling* them under translations. More precisely, the problem asks to find a translation vector $t \in \mathbb{R}^d$ of Q that minimizes the volume or the surface area of the convex hull of $P \cup Q_t$ under the restriction that their interiors remain disjoint, where $Q_t = \{q + t \mid q \in Q\}$.

For two convex polygons in the plane, Lee and Woo showed that the area and perimeter can be minimized in $O(n)$ time [10], where n denotes the number of vertices of P and Q. Research towards bundling more than two polygons would be one very natural direction after Lee and Woo. If one allows the problem to take the number of polygons as part of input, it is NP-hard, even if the input polygons are rectangles by a reduction from the Partition problem [6]. Very recently, Ahn et al. [1] considered the bundling problem with three convex polygons with n vertices in total in the plane and showed that the complexity of the configuration space is $O(n^2)$ and an optimal solution can be computed in $O(n^2)$ time.

Another direction of research naturally takes extension towards higher dimension into account, which is of our interest in this paper. To the best of our knowledge, for dimension $d \geq 3$ there is no known exact algorithm that finds a minimum convex set that can contain two given polytopes P and Q under translations without overlap between their interiors. Ahn et al. [2] considered the problem of minimizing the volume of the convex hull of two convex polytopes under translations for dimension $d \geqslant 3$ where the polytopes are allowed to freely overlap. They presented an algorithm that computes the optimal translation in $O(n^{d+1-\frac{3}{d}} \log^{d+1} n)$ expected time, where n is the total complexity of P and Q.

A special case of this problem, called *a packing problem*, has been studied in the literature, where the shape of the container is predetermined. Then the problem becomes to find a minimum size container of the given shape into which input objects can be placed. In most cases, the containers are simple convex figures such as rectangles and circles, and input objects are polygons in the plane. Milenkovic [11] gave a $O(n^{k-1} \log n)$ time algorithm for packing k convex n-gons into a minimum area axis-parallel rectangle. Alt and Hurtado [4] presented a near-linear time algorithm for packing two convex polygons into a minimum area or perimeter rectangle. Sugihara et al. [13] considered a circle container enclosing a set of input disks in the plane, and gave a "shake-and-shrink" algorithm that shakes the disks and shrinks the enclosing circle step by step.

In this paper, we consider the bundling problem of two convex d-polytopes with n vertices in total in dimension $d \geqslant 3$ where input polytopes are restricted to be in contact under translation. We give an algorithm with running time $O(n^3)$ for $d = 3$ to find a translation vector t^* that attains the minimum volume or surface area of the convex hull of $P \cup Q_{t^*}$. Our algorithm constructs an arrangement in our translation space and evaluates the volume or surface area function on each cell of the arrangement. Our approach extends to any fixed dimension $d > 3$, yielding a first exact algorithm with running time $O(n^{d+\lfloor \frac{d}{2} \rfloor (d-3)})$.

2 Preliminaries

For any subset $A \subseteq \mathbb{R}^d$, let $\mathrm{bd}(A)$ be the boundary of A and $\mathrm{conv}(A)$ the convex hull of A. We denote by $|A|$ and $\|A\|$ the surface area and the volume of A, respectively, when both are well defined for A.

*The work by H.-K. Ahn and D. Park was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MSIP) (No. 2011-0030044 and No. 2013K2A2A4003667). The work by S.W. Bae was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2013R1A1A1A05006927).

†Dept. Computer Science and Engineering, POSTECH, South Korea. {heekap,dwpark}@postech.ac.kr

‡Dept. Computer Science, Kyonggi University, South Korea. swbae@kgu.ac.kr

§Dept. Computer Science, KAIST, South Korea. otfried@kaist.edu

¶Dept. Digital and Information Engineering, Hankuk University of Foreign Studies, Yongin, South Korea. cssin@hufs.ac.kr

Let P and Q be convex d-polytopes in $\mathbb{R}^d$ and n denote the number of vertices of P and Q in total. Without loss of generality, we assume that P is stationary and only Q can be translated by vectors $t \in \mathbb{R}^d$. We denote by Q_t the translate of Q by $t \in \mathbb{R}^d$, that is, $Q_t = \{q + t \mid q \in Q\}$.

Let $\mathrm{vol}(t) = \| \mathrm{conv}(P \cup Q_t) \|$ and $\mathrm{surf}(t) = | \mathrm{conv}(P \cup Q_t)|$. Once t is fixed and the description of $\mathrm{conv}(P \cup Q_t)$ is identified, we can evaluate $\mathrm{vol}(t)$ and $\mathrm{surf}(t)$ in time linear to the complexity of $\mathrm{conv}(P \cup Q_t)$.

Ahn et al. [2] showed that the function $\mathrm{vol}(t)$ is convex on the whole domain $\mathbb{R}^d$. The convexity of the function $\mathrm{surf}(t)$ was proved by Ahn and Cheong [3] for 2-dimensional case only, but their argument can easily be extended to higher dimensions by using the Cauchy's surface area formula for a compact convex subset (See Theorem 5.5.2 in [9]).

For our problem where no overlap between two polytopes is allowed, one might conjecture that there should be an optimal solution such that the two polytopes are in contact with each other. Much to our surprise, this is not always the case. Figure 1 illustrates an example of two polytopes P and Q such that their convex hull volume is minimized when they are *apart*. In the example, P is the polytope defined by the common intersection of $\mathrm{conv}(aa' \cup bb')$ and the halfspace $z \leqslant 20 - \epsilon$ for small $\epsilon > 0$, and Q is just the segment bb'. Then, the convex hull of P and Q forms a tetrahedron which is $\mathrm{conv}(aa' \cup bb')$. Imagine that Q translates slightly in $-z$ direction. Then the convex hull loses volume by points leaving the convex hull through the faces $\mathrm{conv}(bb' \cup qq')$ and $\mathrm{conv}(bb' \cup pp')$, but it gains much larger volume by points entering the convex hull through the faces $\mathrm{conv}(\{b\} \cup aa')$ and $\mathrm{conv}(\{b'\} \cup aa')$. Similarly, one can check that the convex hull volume strictly increases if Q translates in any direction from the placement depicted in Figure 1. Note that this construction can be extended to dimensions higher than 3.

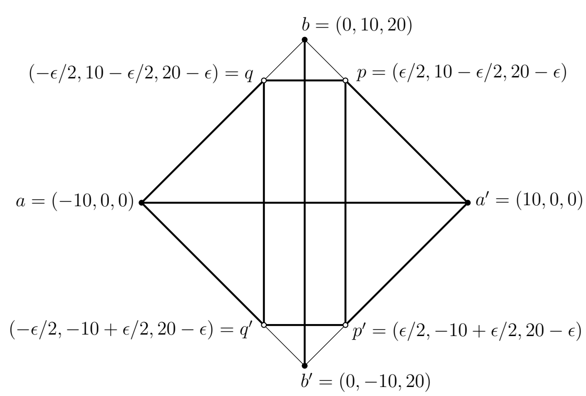

Figure 1: An example of two polytopes P and Q such that the optimal solution make them apart.

As discussed above, the objective functions $\mathrm{vol}(t)$ and $\mathrm{surf}(t)$ are convex over $t \in \mathbb{R}^d$. Thus, if t^* is an optimal solution for our problem without overlap, then either P and Q_{t^*} are apart (in this case, t^* minimizes $\mathrm{vol}(t)$ or $\mathrm{surf}(t)$ over the whole domain $\mathbb{R}^d$), or P and Q_{t^*} are in contact. The former case, which is also the case of Figure 1, can be handled by algorithms for minimizing the volume function $\mathrm{vol}(t)$ when overlap is allowed [2]. While not mentioned in the paper, the same algorithm works for minimizing the surface area function $\mathrm{surf}(t)$.

In this paper, therefore, we focus on the problem where two polytopes are supposed to be in contact with each other. That is, we wish to minimize the volume or the surface area of the convex hull under the restriction that the two polytopes are in contact.

Representing the configuration space. Without loss of generality, we assume that Q contains the origin. Let r be a point of Q that corresponds to the origin, and we call it *the reference point* of Q. Then the translation of Q is specified by the location of the reference point. Imagine that we slide Q along the boundary of P over all possible translations t such that P and Q_t are in contact. Then, the trajectories of r form the boundary of the *Minkowski difference*, denoted by $P \oplus (-Q)$, where $\oplus$ denotes the Minkowski sum and $-Q$ denotes the point reflection of Q with respect to the origin, which is well known in motion planning [7]. Since our polytopes are convex, we have the following lemma.

Lemma 1 *The set of trajectories of the reference point r of Q over all translations t such that P and Q_t are in contact forms the boundary of $P \oplus (-Q)$.*

In our problem, we restrict the two polytopes P and Q to be in contact, and thus the set of all such translations determines the space of all configurations. Lemma 1 suggests that the *configuration space* $\mathcal{K}$ should be defined as the boundary of $P \oplus (-Q)$.

Since P and Q are convex, computing the configuration space $\mathcal{K} = \mathrm{bd}(P \oplus (-Q))$ for P and Q, and consequently specifying all the faces of $\mathcal{K}$ can be done efficiently by a lifting technique, called the *Cayley trick*. This concept starts by introducing the *weighted Minkowski sum* $(1 - \lambda)P_1 \oplus \lambda P_2$ of two convex d-polytopes P_1 and P_2 for $0 \leqslant \lambda \leqslant 1$. The Cayley trick then lifts P_1 and P_2 in a space of one dimension higher with a $(d + 1)$-th coordinate x_{d+1}: P_1 is embedded in the hyperplane $\{x_{d+1} = 0\}$ and P_2 in $\{x_{d+1} = 1\}$. To obtain the weighted Minkowski sum of P_1 and P_2 at any λ, one can just compute the convex hull $\mathrm{conv}(P_1 \cup P_2)$ in $\mathbb{R}^{d+1}$ and cut it through the hyperplane $\{x_{d+1} = \lambda\}$. It is easy to see that the Minkowski sum $P_1 \oplus P_2$ is just a scaled copy of the cut with $\lambda = \frac{1}{2}$. We refer to Huber et al. [8] for more details regarding the Cayley trick.

Note that the convex hull $\mathrm{conv}(P_1 \cup P_2)$ of P_1 and P_2 in $\mathbb{R}^{d+1}$ coincides with the convex hull of the vertices of

P_1 and P_2. Since the complexity of $P_1 \oplus P_2$ does not exceed that of the convex hull $\text{conv}(P_1 \cup P_2)$, we have the upper bound $O((n_1+n_2)^{\lfloor \frac{d+1}{2} \rfloor})$ on the complexity of the Minkowski sum $P_1 \oplus P_2$ of two convex d-polytopes [12], where n_1 and n_2 denote the number of vertices of P_1 and P_2, respectively. Computing $P_1 \oplus P_2$ can be done in $O((n_1+n_2)\log(n_1+n_2)+(n_1+n_2)^{\lfloor \frac{d+1}{2} \rfloor})$ time [5] for any fixed $d \geq 2$. Using this in our configuration space $\mathcal{K}$ yields the following.

Lemma 2 *Let P and Q be convex d-polytopes with n vertices in total for any fixed $d \geqslant 2$. The configuration space $\mathcal{K} = \text{bd}(P \oplus (-Q))$ for P and Q has $O(n^{\lfloor \frac{d+1}{2} \rfloor})$ combinatorial complexity and can be computed in $O(n \log n + n^{\lfloor \frac{d+1}{2} \rfloor})$ time.*

In the following sections, we introduce a decomposition of the configuration space $\mathcal{K}$ and describe a complete algorithm, mainly for dimension $d = 3$. This will lead to a direct extension to higher dimension for $d > 3$.

3 Subdividing the Configuration Space

In this section, we assume $d = 3$. For any translation $t \in \mathcal{K}$, P and Q_t are in contact, more precisely, with contact between a vertex, edge, or facet f of P and a vertex, edge, or facet g of Q. We call the pair (f, g) the *contact pair* at translation $t \in \mathcal{K}$, denoted by $C(t)$. Our approach is to subdivide the configuration space $\mathcal{K}$ into cells so that the contact pair and the convex hull structure of the polytopes do not change within each cell. We then obtain an expression for the volume or surface area function for the convex hull in each cell, and compute its minimum.

By Lemmas 1 and 2, we know that the configuration space $\mathcal{K} = \text{bd}(P \oplus (-Q))$ describes all possible translation vectors and can be constructed in $O(n^2)$ time for $d = 3$. In the following, we further investigate the structure of the configuration space $\mathcal{K}$ to understand the correspondence between its faces and the contact pair that corresponds to a translation.

Imagine that Q translates around P over all possible ways, keeping in contact with each other. This motion is piecewise linear: For any face a of P and face b of Q, let $\sigma_{a,b} \subset \mathcal{K}$ denote the set of translations t such that the contact pair $C(t) = (a, b)$. In the following, we discuss only the case where $\sigma_{a,b} \neq \emptyset$. (1) When a is a facet and b is a vertex, $\sigma_{a,b}$ forms a polygon, which is in fact a translate of a. See (f, u) in Figure 2. When a is a vertex and b is a facet, then $\sigma_{a,b}$ forms a polygon which is a translate of the point reflection of b. See (v', g) in Figure 2. More importantly, observe that $\sigma_{a,b} = a \oplus (-b)$ forms a facet (or 2-face) of $\mathcal{K}$. (2) When both a and b are edges, the subset $\sigma_{a,b}$ forms a parallelogram $a \oplus (-b)$ that is a facet of $\mathcal{K}$. See (vv', uu') in Figure 2. (3) When a is a vertex and b is an edge, $\sigma_{a,b}$ forms a line segment that is a translate of $-b$ by translation vector a. When a is an edge and b is a vertex, $\sigma_{a,b}$ forms a line segment that is a translate of a. See (vv', u') in Figure 2. In this case, $\sigma_{a,b}$ forms an edge of $\mathcal{K}$. (4) When both a and b are vertices, $\sigma_{a,b}$ is a point $a - b$, which is a vertex of $\mathcal{K}$. See (v, u) in Figure 2.

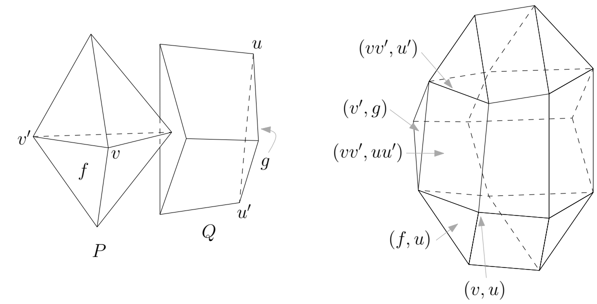

Figure 2: Contact pairs between P and Q, and the configuration space $\mathcal{K}$. Each of vertex-facet pairs, (f, u) and (v', g), defines a facet, an edge-edge pair (vv', uu') defines a facet, a vertex-edge pair (vv', u') defines an edge, and a vertex-vertex pair (v, u) defines a vertex in the configuration space $\mathcal{K}$.

These observations are summarized as follows.

Lemma 3 *Each face (of any dimension) of the configuration space $\mathcal{K}$ corresponds to the set of translations t such that the contact pair $C(t)$ remains the same.*

Hull event planes and horizons. In addition, we have to handle changes in the combinatorial structure of the convex hull $\text{conv}(P \cup Q_t)$ while t continuously varies over $\mathcal{K}$. A change in the structure of the convex hull implies such a motion that a vertex of P and Q either sticks out $\text{conv}(P \cup Q_t)$ from inside or sinks into $\text{conv}(P \cup Q_t)$ from its boundary. In either case, such a change corresponds to a degenerate scene in which Q_t touches the supporting plane of a facet f of P in the same side where P lies. For a facet f of P, consider the set Π_f of all translation vectors $t \in \mathbb{R}^3$ that make such a scene described as above. Since a unique vertex of Q_t must lie on the supporting plane of f for all $t \in \Pi_f$ this set Π_f forms a plane in the space $\mathbb{R}^3$. We then define $h_f := H_f \cap \mathcal{K}$. We call Π_f a *hull event (hyper)plane* and h_f a *hull event horizon.* Each $t \in h_f$ is called a *hull event.* The same discussion also holds for any facet of Q. A proof of the following lemma can be found in the full version of the paper.

Lemma 4 *For any facet f of P or Q, the hull event horizon h_f forms a closed polygonal curve on $\mathcal{K}$ consisting of $O(n^2)$ line segments.*

Now, we consider the subdivision $\mathcal{A}$ of $\mathcal{K}$ induced by h_f for all facets f of P and Q. Observe that for each cell

σ of $\mathcal{A}$, the structure of the convex hull $\text{conv}(P \cup Q_t)$ for all $t \in \sigma$ does not change since we need to cross at least one hull event horizon in order to have a structural change of $\text{conv}(P \cup Q_t)$. Together with the faces of $\mathcal{K}$, $\mathcal{A}$ produces new faces by subdividing faces of $\mathcal{K}$. Since all the hull event horizons are polygonal on $\mathcal{K}$, we regard $\mathcal{A}$ as another convex polytope with parallel facets and edges. Together with Lemma 3, we conclude the following.

Lemma 5 *Let σ be a face of $\mathcal{A}$ (of any dimension). Then, both the contact pair $C(t)$ and the structure of the convex hull $\text{conv}(P \cup Q_t)$ stay constant over all $t \in \sigma$.*

We now bound the complexity of $\mathcal{A}$ with help of the following observation.

Lemma 6 *Any two hull event horizons h_f and h_g for facets f and g cross at most twice.*

Since there are $O(n)$ facets of P and Q in total, Lemmas 4 and 6 imply an immediate upper bound $O(n^3)$ on the complexity of $\mathcal{A}$.

Lemma 7 *The polytope $\mathcal{A}$ consists of $O(n^3)$ faces (vertices, edges, and facets).*

This bound $O(n^3)$ seems easy and improvable, but it is shown to be tight in the worst case.

Tight lower bound construction for $\mathcal{A}$. Figure 3 illustrates an instance of two polytopes which make $\Omega(n)$ closed polygonal curves each of which consists of $\Omega(n^2)$ line segments. Let us describe how to construct two polytopes P and Q more precisely. Figure 3(a) illustrates Q viewed at approximately 7 times magnification. It looks like an "axe" whose head is the segment uu' and blade is the polygonal chain marked as thick segments in the figure. The polytope P is illustrated in Figure 3(b), which can be described as the convex hull of a folding fan with rotating center (pivot) at c and the zigzag edges (thick segments) along its tip. Then we could see that every blade edge constitutes an edge-edge contact pair with each zigzag edge as the blade chain is turning dully. Figure 3(c) shows the configuration space $\mathcal{K}$ for P and Q, which has $\Omega(n^2)$ parallelogram facets corresponding to those edge-edge contact pairs.

Note now that all front facets incident to c have almost the same slope, and all back facets incident to c have almost the same slope. Consider the hull event horizon h_f for a front facet f incident to c. Imagine the motion of Q_t (in the original scale) as t moves along h_f. Then during this motion, the vertex u'' of Q should lie on the supporting plane of f, and each zigzag edge of P sweeps over all the blade edges of Q, resulting in $\Omega(n^2)$ crossings with parallelogram facets of $\mathcal{K}$. See a blue curve in Figure 3(d). Similarly, for any other front and back facet f', the motion of Q_t along $t \in h_{f'}$ results in $\Omega(n^2)$ crossings over the parallelogram facets of $\mathcal{K}$. Therefore, the subdivision $\mathcal{A}$ of $\mathcal{K}$ has $\Omega(n^3)$ complexity.

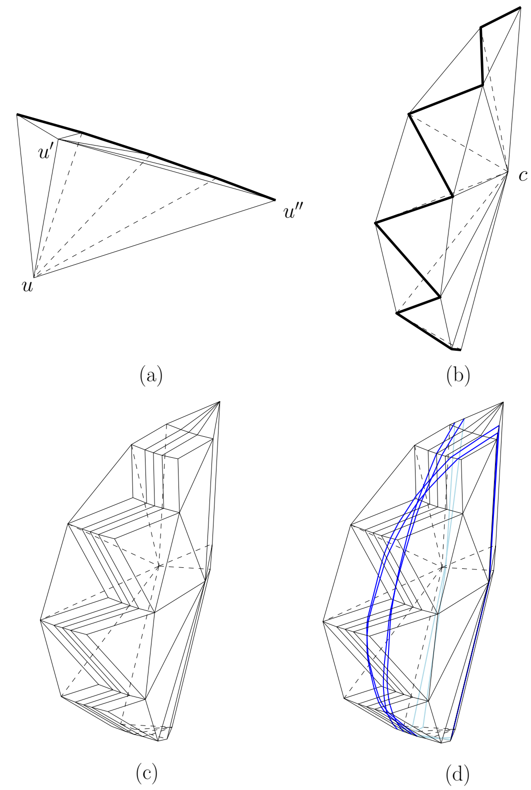

Figure 3: A construction of two polytopes P and Q such that each hull event horizon crosses $\Omega(n^2)$ facets of $\mathcal{K}$. (a) Polytope Q (at 7 times magnification). (b) Polytope P. (c) $P \oplus (-Q)$ whose boundary is $\mathcal{K}$. (d) Four hull event horizons (blue) are drawn on $\mathcal{K}$. Each of them crosses $\Omega(n^2)$ facets of $\mathcal{K}$.

4 Algorithm

In this section, we describe our algorithm, in particular, for dimension $d = 3$ case. Given two convex 3-polytopes P and Q with n vertices in total, our algorithm runs with three stages:

(i) Compute the configuration space $\mathcal{K}$.
(ii) Compute the subdivision $\mathcal{A}$ of the faces of $\mathcal{K}$.
(iii) For each face σ of $\mathcal{A}$, minimize the volume $\text{vol}(t)$ or surface area $\text{surf}(t)$ over $t \in \sigma$.

This basically optimizes our objective function over the whole configuration space $\mathcal{K}$. Thus, the correctness of our algorithm directly follows. In the following, we describe each stage in more details.

Stage (i) can be done by computing the Minkowski sum $P \oplus (-Q)$, which takes $O(n^2)$ time as described in Lemma 2. Recall that $\mathcal{K}$ consists of $O(n^2)$ faces.

In Stage (ii), we repeatedly insert every hull event horizon h_f into $\mathcal{K}$; that is, we cut those faces of $\mathcal{K}$ crossed by h_f and produce new faces. Let $\mathcal{A}_i$ be the resulting subdivision after the i-th insertion of event hull horizon, so $\mathcal{K} = \mathcal{A}_0$ and $\mathcal{A} = \mathcal{A}_m$, where $m = O(n)$ denotes the number of facets of P and Q. At the i-th insertion, let h_f be the horizon to be inserted. We then compute the corresponding hull event plane Π_f and intersect it with $\mathcal{K}_{i-1}$ by tracing h_f and specifying those faces of $\mathcal{K}_{i-1}$ crossed by h_f. This process can be done in time proportional to the number of faces of $\mathcal{K}_{i-1}$ crossed by h_f, which is bounded by $O(n^2 + i)$ by Lemmas 4 and 6. Summing this bound over all $i = 1, \ldots, m$ results in $O(mn^2 + m^2) = O(n^3)$.

Stage (iii) performs actual optimization process for each face σ of $\mathcal{A}$. By Lemma 5, we know that restricting our objective function in each face σ of $\mathcal{A}$ guarantees no change in the contact pair $C(t)$ and the structure of the convex hull over $t \in \sigma$. This means that every vertex of $\mathrm{conv}(P \cup Q_t)$ can be represented by a linear function of t, and $\mathrm{conv}(P \cup Q_t)$ can be triangulated into the same family of tetrahedra in the following way: (1) Triangulate each facet of $\mathrm{conv}(P \cup Q_t)$ if it is not a triangle. (2) choose a point c in the interior of P and connect c to all the vertices of $\mathrm{conv}(P \cup Q_t)$.

Let $\mathcal{T}_\sigma$ be the set of those triangles on $\mathrm{bd}(\mathrm{conv}(P \cup Q_t))$ obtained at Step (1). Also, for each triangle $\triangle \in \mathcal{T}_\sigma$, let $\triangle^+$ be the tetrahedron with base $\triangle$ and apex c. Since P is stationary as assumed, c is fixed and the vertices of each triangle $\triangle \in \mathcal{T}_\sigma$ are linear functions of t on σ, so we write $\triangle(t)$ and $\triangle^+(t)$ as functions of $t \in \sigma$ to denote the geometric triangle and tetrahedron for any fixed $t \in \sigma$. Observe that $\mathrm{vol}(t) = \sum_{\triangle \in \mathcal{T}_\sigma} \|\triangle^+(t)\|$ and $\mathrm{surf}(t) = \sum_{\triangle \in \mathcal{T}_\sigma} |\triangle(t)|$. The volume of a tetrahedron is represented by a cubic polynomial in the coordinates of its vertices, and the area of a triangle by a quadratic polynomial. That is, in a face σ of $\mathcal{A}$, the volume and surface area functions are represented by a polynomial of degree three or two, which can be minimized in $O(1)$ time after having its explicit formula in $O(\mathrm{card}(\mathcal{T}_\sigma)) = O(n)$ time, where $\mathrm{card}(\mathcal{T}_\sigma)$ is the cardinality of $\mathcal{T}_\sigma$. Hence, $O(n)$ time is sufficient for each face of $\mathcal{A}$ to minimize $\mathrm{vol}(t)$ or $\mathrm{surf}(t)$. This implies an $O(n^4)$ time algorithm since $\mathcal{A}$ consists of $O(n^3)$ faces.

Below, we will show that we can do this task in $O(1)$ average time per each face σ of $\mathcal{A}$ by exploiting coherence between adjacent facets.

Exploiting coherence Let σ and σ' be two adjacent facets of $\mathcal{A}$, sharing an edge e. Assume that we have processed σ and we are going to process σ'. We maintain $\mathcal{T}_\sigma$ and all formulas representing $|\triangle(t)|$ and $\|\triangle^+(t)\|$ for each $\triangle \in \mathcal{T}_\sigma$ and their sums (which are $\mathrm{surf}(t)$ and $\mathrm{vol}(t)$). In order to efficiently process the next facet σ', we need to update these invariants. We have two cases here: the edge e is either a portion of an edge of $\mathcal{K}$ or a portion of a hull event horizon h_f for some facet f of P or of Q.

For the former case, we have $\mathcal{T}_{\sigma'} = \mathcal{T}_\sigma$, but the coordinates of the vertices of $\mathrm{conv}(P \cup Q_t)$ should be changed since the contact pair $C(t)$ will be changed by Lemma 3. This causes changes in all formulas for $|\triangle(t)|$ and $\|\triangle^+(t)\|$ for $\triangle \in \mathcal{T}_{\sigma'}$. Thus, in this case, we spend $O(n)$ time because $\mathcal{T}_\sigma$ consists of $O(n)$ triangles.

For the latter case, where e is a portion of a hull event horizon h_f for some facet f of P or Q, σ and σ' belong to a common facet of $\mathcal{K}$. Thus, the contact pair $C(t)$ does not change over $\sigma \cup \sigma'$, while the triangulations $\mathcal{T}_\sigma$ and $\mathcal{T}_{\sigma'}$ differ. Note that for $\triangle \in \mathcal{T}_\sigma \cap \mathcal{T}_{\sigma'}$, the formulas for $|\triangle(t)|$ and $\|\triangle^+(t)\|$ remain the same over $t \in \sigma \cup \sigma'$. Thus, in this case, we are more interested in those triangles $\triangle$ in the symmetric difference between $\mathcal{T}_\sigma$ and $\mathcal{T}_{\sigma'}$, denoted by $\mathcal{T}_e$. Since $e \subset h_f$, for any $t \in e$, the position of P and Q_t implies a degenerate scene such that a vertex u of P or Q lies on the supporting plane of f. As t moves into σ' or into σ, the triangles on f disappear and the triangles determined by each edge incident to f and vertex u appear. This implies that the number of triangles in the symmetric difference $\mathcal{T}_e$ does not exceed twice the number of edges incident to facet f. In order to maintain our invariants, we are done by specifying all appearing and disappearing triangles $\triangle \in \mathcal{T}_e$ and then updating the formulas for the volume or surface area. This can be done in $O(N_f)$ time, where N_f denotes the number of edges incident to f.

To conclude our main result, we need the following lemma, whose proof can be found in the full version of the paper.

Lemma 8 *The total number of triangles in $\mathcal{T}_e$ over all edges e of $\mathcal{A}$ that are portions of some hull event horizon is bounded by $O(n^2 \cdot \sum_f N_f) = O(n^3)$.*

We are now ready to describe Stage (iii) of our algorithm. We traverse all facets of $\mathcal{A}$ from an arbitrary initial facet σ_0. For the first time, we compute $\mathrm{conv}(P \cup Q_t)$ for some $t \in \sigma_0$ and all the invariants from scratch in $O(n^2)$ time. We then minimize our objective function $\mathrm{vol}(t)$ or $\mathrm{surf}(t)$ over $t \in \sigma_0$. As we move on to the next facet σ' from the current facet σ, we update our invariants as described above, according to the type of the edge e between σ and σ', and then minimize the objective function. Repeat this procedure until we traverse all the facets of $\mathcal{A}$.

By a standard traverse, such as the depth first search, we do not cross the same edge more than twice. This implies that the total cost of crossing edges that come from hull event horizons is not more than $O(n^3)$ by Lemma 8. Moreover, if we take a little smarter traverse order, then we can bound the number of crossed edges that are portions of edges of $\mathcal{K}$ by $O(n^2)$. Since each edge crossing of this type costs $O(n)$ time, we finally bound the total cost of update by $O(n^3)$ time.

We finally conclude the following theorem.

Theorem 9 *Given two convex 3-polytopes P and Q with n vertices in total, a minimum convex container bundling P and Q under translations without overlap can be computed in $O(n^3)$ time with respect to volume or surface area.*

5 Extension to Higher Dimensions

Our approach applied to dimension $d = 3$ immediately extends to any fixed higher dimension $d > 3$. In this section, we let $d \geqslant 2$ be any fixed number, and P and Q be two convex d-polytopes with n vertices in total. It is easy to check that Lemma 3 holds for any $d > 3$. As defined for $d = 3$, the hull event hyperplanes Π_f for each facet of P or Q is determined and the intersection $\mathcal{K} \cap h_f$ defined the hull event horizon h_f. Then, the subdivision $\mathcal{A}$ on $\mathcal{K}$ induced by all the hull event horizons possesses the property of Lemma 5.

One important task is to bound the complexity of the subdivision $\mathcal{A}$. A proof of the following lemma can be found in the full version of the paper.

Lemma 10 *For any fixed $d \geqslant 2$, the complexity of the subdivision $\mathcal{A}$ is $O(n^{\lfloor \frac{d}{2} \rfloor (d-3)+d})$.*

Note that the bound for $d = 2$ or 3 in Lemma 10 matches for the previously known upper bounds: Lee and Woo [10] for $d = 2$ and the last sections of this paper for $d = 3$.

Our algorithm for $d = 3$ also extends to any fixed dimension $d > 3$. Stage (i) can be done in $O(n^{\lfloor \frac{d+1}{2} \rfloor})$ time, resulting in the configuration space $\mathcal{K}$ of complexity $O(n^{\lfloor \frac{d+1}{2} \rfloor})$ by Lemmas 1 and 2.

For Stage (ii), there are $O(n^{\lfloor \frac{d}{2} \rfloor})$ facets of d-polytopes P and Q, and thus the same number of hull event horizons on $\mathcal{K}$. As done for $d = 3$, we compute the subdivision $\mathcal{A}$ on $\mathcal{K}$ by adding the hull event horizons one by one. This can be done in time $O(n^{\lfloor \frac{d}{2} \rfloor (d-3)+d})$ by Lemma 10.

Stage (iii) also performs optimization over each facet σ of $\mathcal{A}$ based on the triangulation $\mathcal{T}_\sigma$. In this case, the triangulation $\mathcal{T}_\sigma$ subdivides the boundary of conv$(P \cup Q_t)$ into $(d-1)$-simplices $\triangle$ (i.e., simplices of dimension $d-1$) and for each $\triangle \in \mathcal{T}_\sigma$, we augment one more interior point $c \in P$ to obtain $\triangle^+$ as the d-simplex and thus to triangulate the interior of conv$(P \cup Q_t)$. Note that $\mathcal{T}_\sigma$ consists of at most $O(n^{\lfloor \frac{d}{2} \rfloor})$ $(d-1)$-simplices. The d-dimensional volume of a d-simplex is represented as a polynomial of degree d in the coordinates of its vertices, and so the volume function vol(t) is, while the surface area function surf(t) is represented as a polynomial of degree $d-1$ since it is the sum of the $(d-1)$-dimensional volume of all $\triangle \in \mathcal{T}_\sigma$. By exploiting the coherence among the facets of $\mathcal{A}$ as done for $d = 3$, we can complete Stage (iii) in time $O(n^{\lfloor \frac{d}{2} \rfloor (d-3)+d})$.

We finally conclude the following.

Theorem 11 *For any fixed $d \geqslant 2$ and two convex d-polytopes P and Q with n vertices in total, a minimum convex container bundling P and Q under translations without overlap can be computed in $O(n^{\lfloor \frac{d}{2} \rfloor (d-3)+d})$ time with respect to volume or surface area.*

References

[1] H.-K. Ahn, H. Alt, S. W. Bae, and D. Park. Bundling three convex polygons to minimize area or perimeter. In *Proc. 13th Algorithms and Data Structures Symposium (WADS 2013)*, pages 13–24, 2013.

[2] H.-K. Ahn, P. Brass, and C.-S. Shin. Maximum overlap and minimum convex hull of two convex polyhedra under translations. *Computational Geometry: Theory and Applications*, 40:171–177, 2008.

[3] H.-K. Ahn and O. Cheong. Aligning two convex figures to minimize area or perimeter. *Algorithmica*, 62:464–479, 2012.

[4] H. Alt and F. Hurtado. Packing convex polygons into rectangular boxes. In *Proc. 3rd Japanese Conference on Discrete and Computational Geometry (JCDCG 2000)*, volume 2098 of *LNCS*, pages 67–80, 2001.

[5] B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete & Computational Geometry*, 10:377–409, 1993.

[6] K. Daniels and V. Milenkovic. Multiple translational containment, part i: An approximation algorithm. *Algorithmica*, 19:148–182, 1997.

[7] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 3rd edition, 2008.

[8] B. Huber, J. Rambau, and F. Santos. The Cayley trick, lifting subdivisions and the Bohne–Dress theorem on zonotopal tilings. *Journal of European Mathematical Society*, 2:179–198, 2000.

[9] D. A. Klain and G.-C. Rota. *Introduction to Geometric Probability*. Cambridge University Press, 1997.

[10] H. Lee and T. Woo. Determining in linear time the minimum area convex hull of two polygons. *IIE Transactions*, 20:338–345, 1988.

[11] V. Milenkovic. Translational polygon containment and minimum enclosure using linear programming based restriction. In *Proc. 28th Annual ACM Symposium on Theory of Computation (STOC'96)*, pages 109–118, 1996.

[12] R. Seidel. The upper bound theorem for polytopes: an easy proof of its asymptotic version. *Computational Geometry: Theory and Applications*, 5:115–116, 1995.

[13] K. Sugihara, M. Sawai, H. Sano, D.-S. Kim, and D. Kim. Disk packing for the estimation of the size of a wire bundle. *Japan Journal of Industrial and Applied Mathematics*, 21:259–278, 2004.