Algorithms for Maximizing the Volume of Intersection of Polytopes

Komei Fukuda* Takeaki Uno[1†]

Abstract: In this paper, we address the problem of maximizing the volume of the intersection of polytopes in $\mathbb{R}^d$ by translation. We show that (1) the dth root of the objective function is concave, thus the problem can be solved oracle polynomial time by ellipsoidal method, and (2) the problem can be solved in strongly polynomial time in dimension two even for non-convex polygons.

1 Introduction

Let A and B be two polygons in Euclidean plane where the position of A is fixed and the body B can be translated freely. Then, the intersection of A and B changes as the move. Here we consider the *intersection maximization problem*, that is, to maximize the volume of the intersection by translation. This problem is easy to state, but it appears that it has not been investigated in depth. In fact, this problem has been recently proposed as an open problem at an Oberwolfach workshop by P. Brass [2]. Let us consider a simple example in Figure 1. By shifting horizontally the triangle from left to right, the volume of the intersection of the triangle and the square initially increases as a convex function in the translation variable. Once the right half of the triangle is contained in the square, the volume increase as a concave function. Thus, the volume of the intersection has both features of convex and concave functions. One can also see that the function is continuous, but not differentiable. Consequently, maximization of such functions may not be done in a straightforward manner.

In this paper, we address this problem from the computational point of view. Firstly, we show that the dth root of the objective function is concave for any finite number of d-dimensional convex polytopes. It follows that we can solve the problem in oracle polynomial time where the oracle is the computation of the volume and its gradient. We also show that the hyperplanes spanned by the facets of the polytopes define an arrangement where the objective function is a simple polynomial function in each of its regions. By using these properties, we propose a strongly polynomial time algorithm for the case of two convex polygons in the plane. Its time complexity is $O(n^2 \log^2 n)$ and its space complexity is $O(n)$ if they both have at most n edges. We further propose an enumeration based algorithm for the problem with two non-convex polygons which runs in $O(n^4)$ time and $O(n^2)$ space.

*Institute for Operations Research and Institute of Theoretical Computer Science, ETH Zentrum, CH-8092 Zurich, and IMA/ROSO, EPF Lausanne, CH-1015 Lausanne, Switzerland, e-mail:fukuda@ifor.math.ethz.ch

†National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan, e-mail:uno@nii.jp

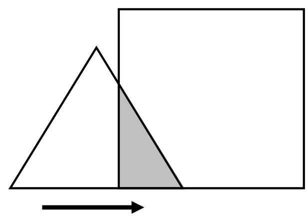

Figure 1: The volume of the intersection of two polygons

2 Preliminaries

We denote the d dimensional Euclidean space by $\mathbb{R}^d$. For a convex body P in $\mathbb{R}^d$ (i.e. a compact convex set with nonempty interior), we denote its volume by $vol(P)$. For a vector h in $\mathbb{R}^d$, $P + h$ is the convex body obtained by translating P by h, i.e., $P + h = \{x + h | x \in P\}$. Whenever there is no confusion, we call a face of $B+h$ face of B. For a family $\mathcal{P}$ of convex bodies $P_1, \ldots, P_k$, we denote its intersection by $\cap(\mathcal{P})$, i.e., $\cap(\mathcal{P}) = \cap_{i=1}^k P_i$.

For a given set of polytopes $\mathcal{P} = \{P_1, \ldots, P_k\}$ in $\mathbb{R}^d$, the intersection maximization problem is to maximize $vol(\cap(\mathcal{P} = \{P_1, P_2 + h_2, \ldots, P_k + h_k\}))$ subject to $h_i \in \mathbb{R}^d$ for $i = 2, \ldots, k$.

3 Convexity on the Intersection Volume of Polytopes

Our first result is the following theorem.

Theorem 1 *For any two convex bodies A and B in $\mathbb{R}^d$, the function $(vol(\cap(\{A, B+h\})))^{1/d}$ is concave in h over the region of nonempty intersection, $\Omega = \{h \in \mathbb{R}^d | \cap (\{A, B + h\}) \neq \emptyset\}$.*

To prove the theorem, we use the Brunn-Minkowski theorem below. Let us denote by $H(z, d)$ the hyperplane in $\mathbb{R}^d$ given by $x_d = z$.

Theorem 2 (Brunn-Minkowski, see [1]) *For any convex body P in $\mathbb{R}^d$, the function $(vol(\cap(\{P, H(z, d)\})))^{1/(d-1)}$ is concave in z over the region $\Omega = \{z \in \mathbb{R}^d | \cap (\{P, H(z, d)) \neq \emptyset\}$.*

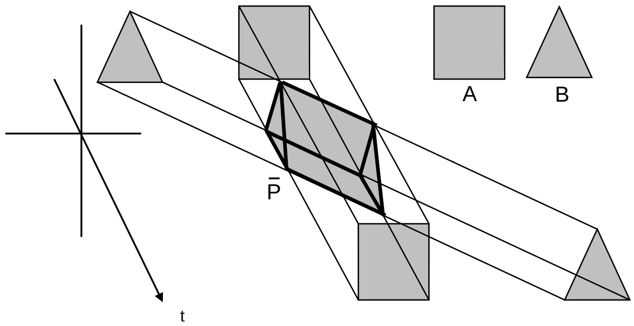

Figure 2: Polytopes $\bar{A}$, $\bar{B}$, and $\bar{P}$

Proof: (of Theorem 1) Let $h, h' \in \mathbb{R}^d$ be any vectors such that both $\cap(\{A, B+h\})$ and $\cap(\{A, B+h'\})$ are non-empty. It suffices to show that the function is concave over the line segment connecting h and h'. Let

$$\bar{A} = \{(x_1,\dots,x_d,\lambda) \mid 0 \le \lambda \le 1, (x_1,\dots,x_d) \in A\},$$
$$\bar{B} = \{(x_1,\dots,x_d,\lambda) \mid 0 \le \lambda \le 1, (x_1,\dots,x_d) \in B + (\lambda h + (1-\lambda)h'))\}.$$

Since both sets are convex, $\bar{P} = \bar{A} \cap \bar{B}$ is also convex, see Figure 2. Consequently, the intersection of $H(\lambda, d+1)$ and $\bar{P}$ is

$$\begin{aligned} H(\lambda,&d+1) \cap \bar{P} \\ &= \{(x_1,\dots,x_d,\lambda) \mid (x_1,\dots,x_d) \in A, \\ &\quad (x_1,\dots,x_d) \in B + (\lambda h + (1-\lambda)h')\}, \\ &= \{(x_1,\dots,x_d,\lambda) \\ &\quad (x_1,\dots,x_d) \in A \cap B + (\lambda h + (1-\lambda)h')\}. \end{aligned}$$

Thus, the intersection is a lifted copy of $\cap(\{A, B + (\lambda h + (1-\lambda)h')\})$. The theorem then follows directly from the Brunn-Minkowski Theorem. ▮

The theorem above can be easily extended to the intersection of several polytopes.

Theorem 3 *For any convex bodies $P_1, \dots P_m$ in $\mathbb{R}^d$, the function $(vol(\cap(P_1+h_1, P_2+h_2, \dots P_m+h_m)))^{1/d}$ is concave in $h = (h_1, \dots, h_m)$ over $\Omega = \{h \mid \cap(P_1 + h_1, P_2 + h_2, \dots P_m + h_m)) \neq \emptyset\}$.*

It follows from Theorem 1 that any locally maximum solution is a global maximum solution. This also implies that the volume of the intersection is a unimodal (quasiconcave) function. Moreover, it is semistrictly quasiconcave. A function $f(x)$ is called *semistrictly quasiconcave* if $f(y) < f(\lambda x + (1-\lambda)y)$ holds for any x and y with $f(x) > f(y)$, and $0 < \lambda < 1$.

For a given point x and a function f having a maximum solution, a hyperplane is called separating hyperplane if it separates x from the set of maximum solutions. Any function can be maximized by ellipsoidal method with finding polynomially many separating hyperplanes. A separating hyperplane of a non-differentiable concave function can be obtained from its subgradient, thus we obtain the following theorem.

Theorem 4 *The problem of maximizing the intersection of d-dimensional polytopes by translation without rotation can be solved in oracle polynomial time in the input size, where the oracle is to compute the volume and its subgradient of the intersection.*

In the next section, we analyze the structure of the objective function so that we can construct a combinatorial algorithm that terminates in strongly polynomial time when $d = m = 2$.

4 Decomposing the Domain into Regions of Equivalence

Let A and B be two convex polygons in $\mathbb{R}^2$, and h be a vector in $\mathbb{R}^2$. Here we regard h as variables. For a vertex v of $\cap(\{A, B+h\})$, we define its *topological representation* by the set of facets (edges) of A and B containing v. The position of v is given by the intersection of these facets, and thus we can represent the position of v as a unique solution to the linear system given by the facets. The solution is a linear function of h. We call this the *functional representation* of v.

Suppose that we are given functional representations of the vertices of $\cap(\{A, B+h\})$, and consider its volume. In general, the volume of a polytope in $\mathbb{R}^d$ is the sum of the volumes of simplexes in a triangulation of the polytope. The volume of each simplex is obtained from the determinant of a matrix consisting of its vertices properly lifted. Therefore, the volume is a polynomial of degree at most d. In the plane, the volume is a quadratic function in h. We call this function the *volume function*.

The following lemma is important.

Lemma 5 *The volume function of $\cap(\{A, B+h\})$ in h and that of $\cap(\{A, B+h'\})$ in h' are identical if the topological representations of the vertices are the same.*

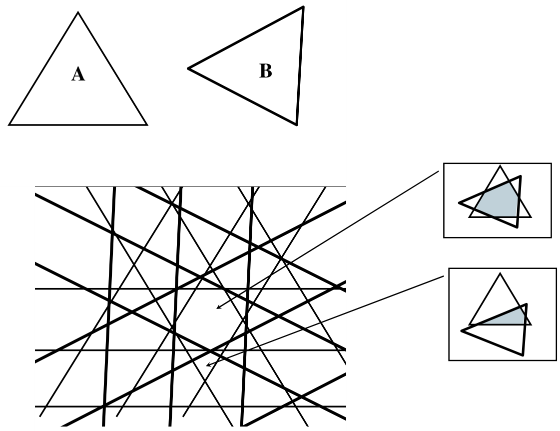

Figure 3: Arrangements by the hyperplanes induced by the facets of A and B

The translations h that induce the same topological representations form an equivalence class. Each equivalence class is determined by a set of linear inequalities which represent conditions such as "a vertex of A (resp., $B+h$) is in a half space induced by a facet of $B+h$ (resp., A)." We denote the set of the hyperplanes (lines) being the boundary of an equivalence class by $\mathcal{H}(\{A,B\})$. The cardinality of $\mathcal{H}(\{A,B\})$ is $O(n^2)$.

Consider the arrangement by the hyperplanes in $\mathcal{H}(\{A,B\})$ (see Figure 3). Then, in a region (i.e a full-dimensional cell) of the arrangement, the set of the topological representations of the vertices does not change. Thus, in each equivalence region, the intersection maximization problem is just a non-convex non-concave quadratic programming with two variables. It can be solved by evaluating the values of the objective function on the vertices, the edges, and the points satisfying that the gradient is 0. It can be done in linear time in the number of edges of the region. Enumerating all regions in the arrangement by $O(n^2)$ lines takes $O(n^4)$ time[3]. The volume function of a region can be computed in constant time from the volume function of a region adjacent to it. Consequently, we can solve the intersection maximizing problem in $O(n^4)$ time and $O(n^2)$ space. Furthermore, this method does not depend on the convexity of two polygons. Therefore, we obtain the following theorem.

Theorem 6 *The problem of maximizing the volume of the intersection of two non-convex polygons in the plane can be solved in $O(n^4)$ time and $O(n^2)$ space.*

Note that one can extend the theorem to general d dimension with k non-convex polytopes. However, the problem is no longer easy, since the volume function is a polynomial of higher degrees, and the number of the regions is huge, exponential in both k and d.

5 Binary Search Algorithm

We here assume that the objective function is perturbed so that the optimal solution is unique. Suppose that H is a hyperplane in $\mathcal{H}(\{A,B\})$, and h is the point in H maximizing $vol(\cap(\{A,B+h\}))$. Then, by looking at the gradient of $vol(\cap(\{A,B+h\}))$, we can check which half space induced by H contains an optimal solution. Using this fact, one can construct a binary search algorithm.

Every hyperplane in $\mathcal{H}(\{A,B\})$ is a translation of a hyperplane spanned by a facet of A or B. It follows that the hyperplanes in $\mathcal{H}(\{A,B\})$ can be partitioned into groups $\mathcal{H}_1,\ldots,\mathcal{H}_{2n}$ such that each group is composed of hyperplanes parallel to each other. Suppose that $\mathcal{H}_i=\{H_1,\ldots,H_k\}$ are sorted in their direction. Then, there is H_j such that an optimal solution is in the area between H_j and H_{j+1}. We call the area between H_j and H_{j+1} the *optimal slab* of $\mathcal{H}_i$. The intersection of the optimal slabs of all $\mathcal{H}_i$ gives the region in which an optimal solution is contained. The optimal slab of $\mathcal{H}_i$ can be found by binary search on $\mathcal{H}_i$, by solving $O(\log n)$ intersection maximization problems on a hyperplane.

Although a straightforward binary search on a line needs $O(n^2)$ preprocessing time and $O(n^2)$ memory, one can reduce them to $O(n\log n)$ time and $O(n)$ space by a median finding like algorithm and implicit representation of $\mathcal{H}_i$.

6 Solving the Problem on a Line

For a hyperplane (line) $H\in\mathcal{H}_i$, we present a method to find an optimal solution h^* in H, which maximizes the volume function restricted to H. Since a line is partitioned into intervals by hyperplanes in $\mathcal{H}(\{A,B\})$, our objective is to find the *optimal interval*, i.e., the interval containing the optimal solution.

Let $x_1,\ldots,x_m$ be the intersection points of H_i and hyperplanes in $\mathcal{H}(\{A,B\})$ which are not parallel to

H. If $x_1, \ldots, x_m$ are sorted in the order of their positions on H, we can easily perform binary search in $O(n \log n)$ time. However, to compute all $x_1, \ldots, x_m$, we need $O(n^2)$ time. Moreover, we need $O(n^2)$ space to keep $\mathcal{H}(\{A, B\})$ and $x_1, \ldots, x_m$ in memory. Thus, here we consider how to execute binary search without computing $x_1, \ldots, x_m$.

The basic idea of the binary search is as follows. Suppose that the groups of hyperplanes are $\mathcal{H}_1, \ldots, \mathcal{H}_p$, and the hyperplanes in each $\mathcal{H}_i$ are sorted. Here we do not include the hyperplanes parallel to H in them. We denote by H_i^* the median of $\mathcal{H}_i$ in the sorted order, and the intersection point of H and H_i^* by x_i^*. Then, we find H_z^* such that x_z^* is "middle" of $x_1^*, \ldots, x_p^*$. Here the middle means that z satisfies

$$\sum_{i=1}^{z} |\mathcal{H}_i| \geq (\sum_{i}^{p} |\mathcal{H}_i|)/2, \text{and}$$
$$\sum_{i=z}^{p} |\mathcal{H}_i| \geq (\sum_{i}^{p} |\mathcal{H}_i|)/2.$$

Note that $\sum_i^p |\mathcal{H}_i|$ is the number of intersection points generated by H and hyperplanes in $\mathcal{H}_1, \ldots, \mathcal{H}_p$. Thus, by looking at the gradient at x^*, we can determine at least 1/4 of the hyperplanes which do not give the endpoints of the optimal interval. More precisely, for each $\mathcal{H}_i = \{H_1, \ldots, H_l\}$, suppose that x_i^* is on the side without the optimal solution. The both endpoints of the optimal interval are in the other side, thereby not given by $H_1, \ldots, H_i^*$ (or $H_i^*, \ldots, H_l$). Therefore, we can reduce such $\mathcal{H}_i$ to $\mathcal{H}_i \setminus \{H_1, \ldots, H_i^*\}$. By repeating this process, $\sum_i^p |\mathcal{H}_i|$ decreases by a constant factor each time. This yields a binary search with $O(\log n)$ steps.

Finding x_z^* can be done in $O(n)$ time by a median finding like method. We find the median, usual meaning of median, of $x_1, \ldots, x_p$ by a median finding algorithm in $O(n)$ time. Then we can see that at least a half of $x_1, \ldots, x_p$ will not be x_z^*. In this way, we can iteratively reduce the candidates, and find x_z^* in $O(n)$ time.

The last remaining problem is to reduce $O(n^2)$ space to store each sorted $\mathcal{H}_i$ in memory. For this, we divide $\mathcal{H}_i$ into two groups, and keep them in memory implicitly with $O(1)$ space. Let $v_1, \ldots, v_n$ be the vertices of A sorted in the clockwise order. For an edge e in B, we denote one of its orthogonal direction by $D(e)$. Let v^* and v^{**} be vertices which maximizes and minimizes $D(e)$, respectively. We divide $v_1, \ldots, v_n$ into two groups such that one group is composed of vertices from v^* to v^{**}, and the other is composed of vertices from v^{**} to v^*. We consider that $v_1, \ldots, v_n$ is a cyclic sequence. Then, we can see that the order of the hyperplanes induced by e and vertices from v^* to v^{**} is a sorted order, and that by vertices from v^{**} to v^* is also a sorted order. Thus, keeping these by two end vertices, we can implicitly keep them in memory with $O(1)$ space.

Now we describe the whole algorithm.

Algorithm MaxIntersection (A ,B)
1. sort the vertices of A and B in the clockwise order, respectively
2. **for each** edge e in A and B
3. compute v^* and v^{**}
4. make implicit representation of groups, e and vertices from v^* to v^{**}, and e and vertices from v^{**} to v^*
5. **endfor**
6. **for each** group $\mathcal{H}_i = \{H_1, \ldots, H_l\}$ //binary search for the optimal slab
7. p :=1, $q := k$
8. **while** $p + 1 < q$
9. find the optimal solution h^* in $H_{(p+q)/2}$ by binary search
10. set $p := (p + q)/2$ or set $q := (p + q)/2$ according to the gradient at h^*
11. **endwhile**
12. **endfor**
13. **output** the optimal solution in the intersection of optimal slabs

Finally, we have the following theorem.

Theorem 7 *The maximization problem of the volume of two convex polygons can be solved in $O(n^2 \log^2 n)$ time and $O(n)$ space.* ∎

Acknowledgment

We thank Professor Masakazu Kojima, Akiko Takeda of Tokyo Institute of Technology, Akiyoshi Shioura of Tohoku University, and Junya Goto of Tsukuba University for their answering out questions. This research is partially supported by Swiss National Science Foundation and Japan Society for the Promotion of Science.

References

[1] K. Ball, "An Elementary Introduction to Modern Convex Geometry," *Flavors of Geometry, Math. Sci. Res. Inst. Publ.*, **31**, Cambridge Univ. Press, Cambridge, 1997.

[2] P. Brass, "A Lower Bound for Lebesgue's Universal Cover Problem," *Workshop of Discrete Geometry*, Mathematisches Forschungsinstitut Oberwolfach, 2005.

[3] H. Edelsbrunner and L. J. Guibas, "Topologically Sweeping an Arrangement," *J. Comput., Syst. Sci.*, **38**, 165-194, 1989.

[4] G. T. Toussaint, "A Simple Linear Algorithm for Intersecting Convex Polygons," *The Visual Computer* **1**, 118-123, 1985.