Computation of Certain Measures of Proximity Between Convex Polytopes: A Complexity Viewpoint

N. K. Sancheti and S. S. Keerthi

Department of Computer Science & Automation
Indian Institute of Science, Bangalore-560 012, India

Abstract

Four problems of proximity between two convex polytopes in $\mathbb{R}^d$ are considered. The convex polytopes are represented as convex hulls of finite sets of points. Let the total number of points in the two finite sets be $n$. We show that three of the proximity problems, viz., checking intersection, checking whether the polytopes are just touching and finding the distance between them, can be solved in $O(n)$ time for fixed $s$ and in polynomial time for varying $s$. We also show that the fourth proximity problem of finding the intensity of collision for varying $s$ is NP-complete.

1 Introduction

In applications such as robot motion planning and VLSI layout, which involve the relative motion or placement of several objects, the quantification of proximity between a pair of objects plays an important role. Let $A$ and $B$ be two objects (compact, connected sets in $\mathbb{R}^d$). There are four important proximity problems whose solutions adequately describe the proximity between $A$ and $B$. They are:

$P_1$: check whether $A$ and $B$ intersect;

$P_2$: check if $A$ and $B$ are just touching;

$P_3$: if $A$ and $B$ are non-intersecting, compute the minimum distance between them; and,

$P_4$: if $A$ and $B$ are intersecting, compute the intensity of collision between them.

The problems will be formally defined in the next section.

Let us briefly mention the usefulness of these problems in Robotics. Gilbert and Johnson [11], and Gilbert and Hong [10] use the euclidean distance between the obstacle and the robot parts in collision-avoidance path planning and collision detection. Distance is used as a measure of how far a robot part is from collision with an obstacle. When the two objects intersect the distance between them is zero. This gives no information about the intensity of collision. The negative of the minimum euclidean distance by which the two objects must be relatively translated so as to separate is defined as the intensity of collision.

Buckley [1] uses this measure in a penalty approach to collision-avoidance robot motion planning. Problems $P_1$ and $P_2$ are directly related to the distance and the intensity of collision. Hence they are also useful.

In this paper we discuss the complexity of solving the four problems. The representation for the objects, $A$ and $B$ plays an important role in deciding the complexities. In the literature the two objects are usually taken to be convex. If an object is non-convex it can be usually expressed as the union of a finite number of convex parts. Then, given two objects, the solution of the four problems for each pair of convex parts of the two objects usually yields a solution of the problems for the given pair of objects. If $A$ and $B$ have a smooth boundary then the problems may not be solvable using a finite amount of computation [13]. In this paper we take $A$ and $B$ to be convex polytopes, a representation popularly used in Robotics. A polytope can be described in two ways: (1) the point description, in which the polytope is given to be the convex hull of a finite number of points; and (2) the facial description, in which the polytope is given as the set of points satisfying a finite number of linear inequalities. Most of the results in this paper concern the point description. (A few comments will be made at the end of the paper about complexities associated with the facial description.) Let $m$ and $n$ be integers satisfying $1 \leq m < n < \infty$.

$$A = \text{Co}(A)\quad \text{and} \quad B = \text{Co}(B), \quad (1.1)$$

where $\text{card}(A) = m$, $\text{card}(B) = n - m$, and, given $X \subseteq \mathbb{R}^d$, $\text{Co}(X)$ and $\text{card}(X)$ denote the convex hull of $X$ and the cardinality of $X$, respectively.

We will analyze algorithms for the four problems in terms of two complexity types, described as follows.

- **Type 1 Complexity**: $s$ is assumed to be fixed and, complexity is viewed as a function of $n$ and is measured in terms of the number of arithmetic operations required.

- **Type 2 Complexity**: $s$ is taken as a variable, all data (i.e., $A$ and $B$) are assumed to be rational and, complexity is viewed as a function of $L$, the total data size in number of bits, and measured in terms of the number of bit operations required.

Type 1 complexity is appropriate in applications such as Robotics where $s$ is small, say $s \leq 3$. Type 2 com-
plexity is useful in other applications which involve larger values for $s$.

There is a wealth of literature on the problems, $P_1$ and $P_2$. There are two different approaches. The first approach, with its roots in Computational Geometry, aims to design algorithms with a low type 1 complexity. For $s = 2$, with $A_p$ and $B_p$ given such that their members are the vertices of $A$ and $B$ in order, Chazelle and Dobkin [4] have given an $O(n \log n)$ algorithm for $P_1$ and, Chin and Wang [5] and Edelsbrunner [6] have independently proposed an $O(n \log n)$ algorithm for $P_2$. For $s = 3$, with the complete vertex-edge-face descriptions of $A$ and $B$ given, note that Euler's formula for three dimensional polytopes implies that the size of this description is $O(n)$. Chazelle and Dobkin [4] have given an $O(n)$ algorithm for $P_1$, and, Dobkin and Kirkpatrick [7] have proposed an $O(n)$ algorithm for $P_2$. Megiddo [17, 9] has given an $O(n)$ algorithm for $P_3$ assuming only point descriptions for $A$ and $B$. His algorithm does not restrict itself to $s = 3$. Thus Megiddo's result on the complexity of $P_3$ is a clear improvement of Chazelle and Dobkin's result. Furthermore, Megiddo's ideas can be used to show that $P_4$ has a polynomial type 2 complexity.

The second approach to the solution of $P_1$ and $P_2$, considered by Gilbert et al. [12], has its roots in Mathematical Programming. The main aim of this approach is to design practically efficient algorithms. The algorithm suggested in [2] solves $P_1$ and $P_2$ simultaneously. Although it uses a severe worst case complexity, detailed computational tests have shown that its complexity type 1 complexity is $O(n)$.

Let us now come to the main contributions of this paper. We propose a new approach to the solution of $P_3$ and $P_4$ which can be viewed as a nontrivial extension of Megiddo's approach to $P_1$. Let $A$ and $B$ be given by (1.1). We first reduce $P_3$ to a nice quadratic programming problem. This reduction is combined with two recent quadratic programming algorithms [16, 17, 19] to give: (i) an algorithm for $P_3$ whose type 1 complexity is $O(n)$; and (ii) an algorithm for $P_3$ whose type 2 complexity is polynomial. The former algorithm implies a complexity result which is a clear improvement of Dobkin and Kirkpatrick's result for $P_3$ since: (1) our algorithm is not restricted to $s = 3$; and (2) even for $s = 3$, our algorithm does not require the vertex-edge-face descriptions of $A$ and $B$. (Note that, for $s = 3$, the determination of the vertex-edge-face descriptions of $A$ and $B$ from the point descriptions of $A$ and $B$ requires $O(n \log n)$ effort.) These improvements are similar to those achieved by Megiddo's algorithm over Chazelle and Dobkin's algorithm for solving $P_1$.

Reducing $P_2$ to a problem of checking the feasibility of linear inequalities, we also derive two algorithms for $P_2$ similar to those mentioned above for $P_3$. It appears that, for $P_1$, $P_2$ and $P_3$, $O(n)$ is the optimal type 1 complexity. As already remarked, the type 2 complexity of these problems is polynomial. We briefly remark, towards the end of the paper that, if, facial description of $A$ and $B$ are given and $n$ denotes the total number of linear inequalities, then algorithms can be given to solve $P_1$, $P_2$ and $P_3$ with $O(n)$ type 1 complexity and polynomial type 2 complexity.

The fourth proximity problem of computing the intensity of collision has not received much attention. The only algorithms known for this problem are those of Buckley and Leifer [2], Cameron and Culley [3] and Kehrthi and Sridharan [15]. For $s = 3$ these algorithms have an $O(n^2 \log n)$ type 1 complexity and for variable $s$ they have an exponential type 2 complexity. For $P_4$ our main contribution is the following. We reduce $P_4$ to a problem of maximizing a convex quadratic function subject to linear inequalities and show that, in terms of type 2 complexity, $P_4$ is NP-complete.

The following notations will be used. For $z \in \mathbb{R}^n$, $z^T = \text{Ch term component of } z \text{ and } \|z\| = \text{Euclidean norm of } z$. $A - B$ will denote the inner product of $A$ and $B$. If $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$, $(x, y)$ will denote the joint vector, $(x^T y^T)^T$ where prime denotes transpose. $A - B$ is the Carathéodory-Minkowski set difference between $A$ and $B$, i.e., $A - B = \{z : z = x - y \text{ for some } x \in A, y \in B\}$. We will denote the boundary of set $A$ by $\text{bd}(A)$.

2 Preliminaries

In this section we give formal definitions of the proximity measures between convex polytopes and also define the four proximity problems formally. We will assume the two objects $A$ and $B$ to be convex polytopes for further discussions.

Definition 2.1. The (euclidean) distance between $A$ and $B$, $D_+$ is given by

$$D_+ = \min \{ \|x - y\| : x \in A, y \in B \}.$$ 

Note that when $A \cap B \neq \phi$, $D_+ = 0$. Under such circumstances, it is useful to talk in terms of the intensity of collision. When $A \cap B = \phi$, there is no collision and hence the intensity of collision is taken as zero. A good measure of the intensity of collision should reflect on how close $A$ and $B$ are from separation, i.e., the least relative translation needed for separating $A$ and $B$. To describe relative motion, it is sufficient to consider one of the objects (say, $A$) to be static and other ($B$) to be moving. Let us first concentrate on the translation of $B$ by $x \in \mathbb{R}^n$. Given $x \in \mathbb{R}^n$, let

$$B(z) = B + (z) = \{z + x : z \in B \}.$$ 

Definition 2.2. The intensity of collision between $A$ and $B$, $D_-$ is given by

$$D_- = \inf \{ \|x\| : A \cap B(z) = \phi \}.$$ 

Note that these definitions of $D_+$ and $D_-$ are applicable also to the case when $A$ and $B$ are general convex objects (not necessarily convex polytopes). Let us now define the concept of $A$ and $B$ just touching each other.

Definition 2.3. $A$ and $B$ are said to be just touching if $A \cap B \neq \phi$ and $3 \mu \in \mathbb{R}^n$ such that $Q \in > A \cap B(z(\mu)) = \phi$.

The above definition is also applicable when $A$ and $B$ are general convex objects. For nonconvex objects only Definition 2.1 is useful as a proximity measure. The following lemma contains useful results about
The following three lemmas. The ideas for The reductions of problems yields efficient algorithms for well known efficient algorithm.

Lemma 2.1. (i) \(D_1 \geq 0, D_2 \leq 0\); (ii) \(A \cap B \neq \emptyset\) if \(D_1 = D_2 = 0\); and, (iv) \(D_3 = -\sup \{ x : S(x) \subseteq A - B \}\), where \(S(r) = \{ x : \| x \| \leq r \}\), the ball of radius \(r\) centred at origin.

Problems \(P_1 - P_4\) can now be formally defined as

\(P_1\): check if \(A \cap B = \emptyset\);
\(P_2\): check if \(A\) and \(B\) are just touching;
\(P_3\): if \(A \cap B = \emptyset\), find \(D_+\); and,
\(P_4\): if \(A \cap B \neq \emptyset\) find \(D_-\).

We define the support functions of the objects, \(h_1, h_2\) as

\[ h_1(\eta) = \min \{ \eta \cdot x : x \in A \}, \]

and,

\[ h_2(\eta) = \max \{ \eta \cdot x : x \in B \}. \]

Define \(h : R^t \rightarrow R\) as,

\[ h(\eta) = h_1(\eta) - h_2(\eta). \]

The following lemma contains a few useful properties of \(h_1, h_2\) and \(h\). We omit its proof as it is easy to establish.

Lemma 2.2. (i) \(h_1\) is concave, \(h_2\) is convex and \(h\) is concave; (ii) \(h(\eta) = \min \{ \eta \cdot x : x \in A - B \}\); (iii) \(h_1(\alpha \eta) = \alpha h_1(\eta), h_2(\alpha \eta) = \alpha h_2(\eta)\), and \(h(\eta) = \alpha h(\eta) \forall \alpha \geq 0\); (iv) furthermore, if \(A\) and \(B\) are given by (1.1) then \(h_1(\eta) = \min \{ \eta \cdot v : v \in A_+ \}\) and \(h_2(\eta) = \max \{ \eta \cdot v : v \in B_+ \}\).

3 Solutions of \(P_1, P_2\) and \(P_3\)

In this section we reduce each of the problems, \(P_1\) and \(P_2\), to a problem of checking the feasibility of linear inequalities, and, problem \(P_3\) to a convex quadratic programming problem (QP). Then, the application of well known efficient algorithms for feasibility and QP yields efficient algorithms for \(P_1 - P_3\). Throughout this section and the next we assume the following:

\[ A = \{ v_1, v_2, \ldots, v_m \}; B = \{ v_{m+1}, v_{m+2}, \ldots, v_{n} \}; \]

\[ A = \mbox{Co}(A_+); B = \mbox{Co}(B_+); \]

\[ E = \{ (\eta, c) \in R^{t+1} : \eta \cdot v_i - c \geq 1 \forall i = 1, \ldots, m, \]

\[ \eta \cdot v_i - c \leq -1 \forall i = m + 1, \ldots, n \}; \]

and,

\[ F = \{ (\eta, c) \in R^{t+1} : \eta \neq 0, \eta \cdot v_i \geq \gamma \forall i = 1, \ldots, m, \]

\[ \eta \cdot v_i \leq c \forall i = m + 1, \ldots, n \}. \]

The reductions of problems \(P_1 - P_3\) are described by the following three lemmas. The ideas for \(P_1\) are somewhat different from those given by Megiddo [17, 9].

**Lemma 3.1.** \(A \cap B = \emptyset \iff E \neq \emptyset\).

**Proof.** By the separating hyperplane theorem \(A \cap B = \emptyset\) iff there exists a hyperplane, \(H = \{ x : \beta \cdot x = \alpha \}\), strictly separating \(A\) and \(B\), where \(\alpha \in R, \beta \in R^t\) and \(\beta \neq 0\). Without loss of generality we can assume that \(\beta \cdot x \geq \alpha \forall x \in A\) and \(\beta \cdot x < \alpha \forall x \in B\). Let

\[ \gamma = \max \{ \beta \cdot v_i : m + 1 \leq i \leq n \}. \]

Clearly \(y < \alpha\) and,

\[ \beta \cdot x \geq \alpha \forall x \in A \quad \text{and} \quad \beta \cdot x \leq \gamma \forall x \in B. \]

Since \(A = \mbox{Co}(A_+)\) and \(B = \mbox{Co}(B_+)\), (3.1) is equivalent to

\[ \beta \cdot v_i \geq \alpha, \quad i = 1, \ldots, m, \]

\[ \beta \cdot v_i \leq \gamma, \quad i = m + 1, \ldots, n. \]

Define \(\rho = (\alpha - \gamma)/2\). We can rewrite (3.2) as,

\[ \beta \cdot v_i - \rho \geq (\alpha - \gamma)/2, \quad i = 1, \ldots, m, \]

\[ \beta \cdot v_i - \rho \leq (\alpha - \gamma)/2, \quad i = m + 1, \ldots, n. \]

Define: \(\eta = 2\rho/(\alpha - \gamma)\); and \(c = 2\rho/(\alpha - \gamma)\). Dividing both sides of each of the inequalities in (3.3) by \((\alpha - \gamma)\), it is easy to see that (3.2) is equivalent to

\[ \eta \cdot v_i - c \geq 1, \quad i = 1, \ldots, m, \]

\[ \eta \cdot v_i - c \leq -1, \quad i = m + 1, \ldots, n. \]

This proves the lemma.

**Lemma 3.2.** Suppose \(A \cap B \neq \emptyset\). Then \(A\) and \(B\) are just touching iff \(F \neq \emptyset\).

**Proof.** Let \(A\) and \(B\) be just touching. Let \(\mu \in R^t\) be as in Definition 2.3. Take a sequence of hyperplanes, \(\{H_i\}, \) where \(H_i = \{ x : \eta_i \cdot x = c_i \}, \| \eta_i \| = 1, \)

\(A \subseteq \{ x : \eta_i \cdot x \geq c_i \}\) and \(H_i\) strictly separates \(A\) and \(B(\mu/4)\). We get a bounded sequence \(\{ (\eta_i, c_i) \}\) in \(R^{t+1}\) which admits a convergent subsequence. Let the subsequence converge to \( (\eta, c)\). It is easy to verify that \( (\alpha, c) \in F\). Thus, \(F \neq \emptyset\).

Now assume \(F \neq \emptyset\). Let \(\eta \in F\). It is easy to verify that \(\forall \epsilon > 0, A \cap B(-\epsilon \eta) = \emptyset\), so that \(A\) and \(B\) are just touching.

**Lemma 3.3.** Suppose \(A \cap B = \emptyset\). Then,

(i) \(D_+ = \max \{ h(\eta) : \| \eta \| = 1, h(\eta) > 0 \} \); and,

(ii) \(D_+ = 2/\min \{ \| h(\eta) \| : h(\eta) - c = 0 \}

\[ h_2(\eta) - c \leq -1 \quad \text{for some } c \in R \} \).

**Proof.** By definition,

\[ D_+ \leq \min \{ \| x - y \| : x \in A, y \in B \} \]

\[ \leq \min \{ \| z \| : z \in A - B \}. \]

Let \(z \in A - B\) solve the minimization problem above. (Existence of \(z\) follows from the compactness of \(A - B\) and Weierstrass' theorem.) Define \(\eta = z/\| z \|\). (Note that \(\| z \| = D_+ > 0\).) Optimality of \(z\) implies that the
hyperplane, $\{z : \bar{f} \cdot z = D_+\}$ contains $\bar{z}$ and strictly separates $A - B$ from the origin. In other words,

$$||\bar{z}|| = D_+ = h(\eta).$$  \hfill (3.5)

Now choose any $\eta$ satisfying $||\eta|| = 1$. By part (ii) of Lemma 2.2 and compactness of $A - B$ there exists $w \in A - B$ that satisfies $h(\eta) = \eta \cdot w$. Then, by part (ii) of Lemma 2.2, Schwarz inequality and (3.5) we get

$$h(\eta) = \eta \cdot w \leq ||\eta|| \cdot ||w|| \leq ||\bar{z}|| = D_+ = h(\eta).$$

Thus $D_+ = \max \{h(\eta)/||w|| : h(\eta) > 0\}$. Since $D_+ > 0$, part (i) follows.

Using part (ii) of Lemma 2.2 and part (i) just proved, it is not difficult to see that

$$D_+ = \max \{|h(\alpha)/||\alpha|| : h(\alpha) > 0, \alpha \neq 0\}.$$  \hfill (*)

For a particular $\eta$, $h(\eta)/||\eta||$ and $h(\alpha)/||\alpha||$ are same for $\forall \alpha > 0$. Therefore, we can fix a positive value of $h(\eta)$, say $h(\eta) = 2$. Thus,

$$D_+ = \max \{2||\eta|| : h(\eta) = 2\} = 2/\min \{||\eta|| : h(\eta) = 2\}.$$  \hfill (3.6)

Suppose $h(\eta) > 2$ for some $\eta$. Then $\eta = \alpha \eta_j$, with $
abla \alpha \geq 2/h(\eta)$, satisfies $h(\eta) = 2$ and $||\eta|| < ||\eta||$. Thus,

$$D_+ = 2/\min \{|||\eta|| : h(\eta) \geq 2\}. \hfill (3.8)$$

Also,

$$h(\eta) \geq 2 \iff h(\eta)/||\eta|| \geq 2 \iff h(\eta)/||\eta|| \geq 2 \iff 3 \in \mathbb{R} \text{ such that } h(\eta) - 1 \geq 3 \text{ and } h(\eta) + 1 \leq 3.$$  \hfill (3.9)

Using this in (3.6) proves the lemma.

The above three lemmas yield the reductions of $P_1$- $P_3$ to standard problems. This is stated in the following theorem. Its proof is omitted since it is nearly a restatement of Lemmas 3.1-3.3.

**Theorem 3.4.**

(i) $A \cap B = \emptyset$ iff $3 \eta \in \mathbb{R}^n$ and $c \in \mathbb{R}$ such that

$$(\eta, c) \in E. \hfill (3.7)$$

(ii) $A$ and $B$ are just touching iff $A \cap B \neq \emptyset$ and $3 \eta \in \mathbb{R}^n, c \in \mathbb{R}$, such that

$$(\eta, c) \in F. \hfill (3.8)$$

(iii) If $A \cap B = \emptyset$, $D_+ = 2/\min \{||\eta|| : (\eta, c) \in E\}. \hfill (3.9)$

**Remark 3.1.** In order to solve $P_1$, we have to check the feasibility of (3.7). For this we can use any dummy linear objective function (say, $p^1$) and solve the resulting linear program (LP). This LP is infeasible if $A \cap B \neq \emptyset$. Thus to solve $P_1$, we have to solve one LP involving $(s + 1)$ variables and $n$ inequality constraints.

**Remark 3.2.** Theorem 3.4 shows that $P_3$ also reduces to a feasibility problem. But, we have a non-standard constraint $\eta \neq 0$. We briefly indicate how to solve this problem using a sequence of at most 2s feasibility problems. Consider the system,

$$\eta \cdot v_i - c \leq 0, \quad i = 1, \ldots, m,$$

$$\eta \cdot v_j - c \geq 0, \quad i = m + 1, \ldots, n. \hfill (3.10)$$

The following steps solve $P_2$

0. Set $k = 1$.

1. Fix $\eta^f = 0$ for $j \geq 1 \leq k$ in (3.10) and, let $L^k_j$ and $L^k_{-j}$ denote the systems obtained by further fixing $\eta^f = 1$ and $\eta^f = -1$, respectively. (Essentially, $L^k_j$ and $L^k_{-j}$ are systems involving $s - k$ variables and $n$ inequalities.) Check the feasibilities of $L^k_j$ and $L^k_{-j}$. If one of them is feasible, conclude that $A$ and $B$ are just touching and stop; else, go to step 2.

2. If $k = s$, conclude that $A$ and $B$ are not just touching and stop; else, reset $k = k + 1$ and go back to step 1.

**Remark 3.3.** Solving $P_3$ requires the solution of a convex quadratic programming problem (QP) involving $(s + 1)$ variables and $n$ inequality constraints. Let us now condense some well-known results associated with the solution of the standard problems mentioned in Remarks 3.1-3.3. Consider the QP

$$\min \quad y \cdot T y + p \cdot y \hfill (3.11)$$

s.t. $q_i \cdot y \leq r_i, \quad i = 1, \ldots, n,$ \hfill (3.12)

where $y \in \mathbb{R}^d, d \leq (s + 1)$, and $T$ is a symmetric positive semidefinite matrix. The feasibility problem, i.e., checking the feasibility of (3.12) forms a part of QP and therefore it is a special case of QP. LP is a special case of QP and it corresponds to setting $T \equiv 0$ in (3.11). It is easy to see that each of the problems mentioned in Remarks 3.1-3.3 is a special case of QP. The following results concerning the complexity of solving QP are well-known.

**Lemma 3.5.** There exists an algorithm with $O(n)$ time complexity that solves QP.

**Lemma 3.6.** There exists an algorithm with polynomial type 2 complexity that solves QP.

For LP, Lemma 3.5 is proved by Megiddo in [17], where he gives a detailed algorithm with $O(n)$ time complexity and mentions that a similar algorithm can be derived for QP. In [18] Megiddo gives the details of his QP algorithm for $d \leq 3$. This algorithm requires somewhat messy modifications of his LP ideas. In [19] we have given an elegant algorithm for QP with $O(n)$ time complexity which is directly along the lines of Megiddo’s LP algorithm. For LP, Lemma 3.6 is due to Khachiyan [14] and, for QP it is due to Kozlov et al. [16].
Combining the reductions given in Theorem 3.4 with Remarks 3.1-3.3 and using Lemmas 3.5-3.6 we get the following result on the complexities of $P_1-P_3$.

Theorem 3.7. For each of the problems, $P_1-P_3$, there exist algorithms with $O(n)$ type 1 complexity and polynomial type 2 complexity.

As far as the worst case complexity analysis is concerned, the above result is promising. We have not implemented any of the algorithms which lead to the complexities mentioned in Theorem 3.7. We expect that, for small $s$ (which is typically the case in Robotics), low type 1 complexity algorithms based on those in [17, 19] will be useful. However, practical usefulness of these algorithms is yet to be demonstrated.

The algorithms need to be implemented and compared against practically efficient algorithms such as those in [12].

4 Complexity of $P_4$

In this section we discuss some complexity issues associated with problem $P_4$. As in section 3, we assume that A and B are given by point descriptions. Our main result is that the intensity of collision computation can be reduced to a concave quadratic programming problem. This reduction follows from the following lemma.

Lemma 4.1. Suppose: A and B not just touching each other; and, $h_1$, $h_2$ and $h$ are defined as in (2.1)-(2.3). Then:

(i) $D_- = \max \{h(\eta) : ||\eta|| = 1, h(\eta) < 0\}$ and,
(ii) $D_- = -2/\max \{||\eta|| : h_1(\eta) - c \leq -1 \text{ and } h_2(\eta) - c \leq 1 \text{ for some } c \in \mathbb{R}\}$.

Proof. From parts (iii) and (iv) of Lemma 2.1, $0 > D_- = \sup \{r : S(r) \subseteq A - B\}$. Let $\tilde{r} = -D_-$. Compactness of $A - B$ implies that $S(\tilde{r}) \subseteq A - B$ and that there exists $z \in \partial(A - B) \cap S(\tilde{r})$. Clearly, $||z|| = -\tilde{r} = -D_- > 0$. Define $\tilde{\eta} = -||z||$. Since $S(\tilde{r}) \subseteq A - B$, any hyperplane which supports $A - B$ at $z$ also supports $S(\tilde{r})$. But, there is a unique hyperplane, $H = \{x : z \cdot x = -\tilde{r}\}$, which supports $S(\tilde{r})$ at $z$. Thus, $H$ also supports $S(\tilde{r})$. This implies that $h(\tilde{\eta}) = D_-$. Now choose any $\eta$ such that $||\eta|| = 1$. Since $S(\tilde{r}) \subseteq A - B = -\eta \in A - B$. Hence,

$$h(\eta) \leq h(\tilde{\eta})$$

Thus $D_- = \max \{h(\eta) : ||\eta|| = 1\}$. Since $||\eta|| = 1$ implies $h(\eta) < 0$, this condition is superfluous and part (i) follows.

Using part (iii) of Lemma 2.2 and part (i) just proved, it is not difficult to see that

$$D_- = \max \{h(\eta)/||\eta|| : h(\eta) < 0\}.$$ 

For a particular $\eta$, $h(\eta)/||\eta||$ and $h(\alpha\eta)/||\alpha\eta||$ are same for all $\alpha > 0$. Therefore, we can fix a negative value of $h(\eta)$, say $h(\eta) = -2$. Thus,

$$D_- = \max \{ -2/||\eta|| : h(\eta) = -2\} = -2/\max \{||\eta|| : h(\eta) = -2\}.$$ 

It is not difficult to see that even if we allow $h(\eta) > 2$ in the maximization problem of the above equation, then maximum will be attained at $h(\eta) = -2$. Hence,

$$D_- = -2/\max \{||\eta|| : h(\eta) = -2\}. \quad (4.1)$$

Now,

$$h(\eta) \geq -2 \Rightarrow h_1(\eta) - h_2(\eta) \geq -2 \Rightarrow h_1(\eta) + 1 \geq h_2(\eta) \Rightarrow 3c \in R$$ such that $h_1(\eta) - c \leq -1$ and $h_2(\eta) - c \leq 1$.

Using this in (4.1) proves the lemma.

Combining the above lemma with part (iv) of Lemma 2.2 we get $D_- = -2/\delta$, where $\delta$ is the solution of the following convex maximization problem.

$$\max \{||\eta||\} \quad \text{s.t.} \quad \eta \cdot v_i \leq \epsilon - 1, \quad i = 1, \ldots, m. \quad (4.2)$$

Vavasis [20] has shown that (4.2) belongs to NP. To show that (4.2) is NP-complete we convert the conjunctive normal SAT problem [6] to an instance of (4.2). Consider a formula with s logical variables $\chi_1, \ldots, \chi_s$, and $M$ clauses. Let $\eta'$ denote a real variable corresponding to $\chi_s$, which takes a value of 1 if $\chi_s$ is true and -1 if $\chi_s$ is false. The SAT problem has a solution iff the maximum value of $||\eta||$ subject to $\eta' \leq 1$, $-\eta' \leq 1 \text{ for } j = 1, \ldots, s$, and the clause constraints is s. Each clause constraint can be equivalently written as $\eta \cdot v_i \leq 1$ where $v_i$ is appropriately defined. For example, the clause $\chi_1 \chi_2 \chi_3 \chi_4$ will translate to the inequality

$$(1 + \eta' + (1 - \eta') + (1 - \eta') > 0.5, i.e., 0.4\eta' + 0.4\eta' - 0.4\eta' \geq 1.$$ 

Thus, after a multiplication of each of the inequalities by a factor of 2, SAT reduces to an instance of

$$\max \{||\eta||\} \quad \text{s.t.} \quad \eta \cdot v_i \leq 2, \quad i = 2, \ldots, n. \quad (4.3)$$

where $n = 2s + M + 1$ and the $v_i$'s are appropriately defined. If we define $v_1 = 1, v_1 = 0$, the origin in $R^n$, then it is easy to conclude from (4.2) and (4.3) (by eliminating $c$ from (4.2)) that the maximum value of the objective function in (4.3) is nothing but $-2/D_-$. Thus we have the following result.

Theorem 4.2. $P_4$ is NP-complete in the sense of type 2 complexity.

5 Complexity for Facial Representations

In sections 3 and 4 we have considered only point descriptions of A and B. In this section we consider facial representations of A and B. Let

$$A = \{x \in R^n : a_i \cdot x \leq b_i, i = 1, \ldots, m\},$$

$$B = \{x \in R^n : a_i \cdot x \geq b_i, i = m + 1, \ldots, n\}.$$ 

Theorem 3.7 holds for this modified representation too. The reasoning is as follows. First, $P_3$ is equivalent to checking the feasibility of the system,

$$a_i \cdot z \leq b_i, \quad i = 1, \ldots, m,$$

$$a_i \cdot z \geq b_i, \quad i = m + 1, \ldots, n. \quad (5.1)$$
$P_1 \cap P_2 = \emptyset$ iff the above system is infeasible. Similar to Remark 3.2, $P_3$ can also be solved using a sequence of feasibility problems. The details can be easily worked out. $P_4$ is equivalent to the solution of the QP

$$\min \|x - y\| \text{ s.t. } (5.1).$$

Whether or not a polynomial type 2 complexity algorithm exists for $P_4$ is an interesting open problem.

6 Conclusion

In this paper we have analyzed two types of complexity for four proximity problems associated with a pair of objects whose point descriptions are given. We have shown that $P_1 - P_3$ can be solved in linear type 1 and polynomial type 2 complexities. We have also shown that $P_3$ is NP-complete.

References


