Applied Mathematics and Computation 429 (2022) 127183

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

A fast and efficient algorithm for determining the connected orthogonal convex hulls

Nguyen Kieu Linh [a,b], Phan Thanh An [b,c,*], Tran Van Hoai [b,c]

[a] Posts and Telecommunications Institute of Technology, Hanoi, Viet Nam
[b] Institute of Mathematical and Computational Sciences, Faculty of Computer Science and Engineering, Faculty of Applied Sciences, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, Ward 14, District 10, Ho Chi Minh City, Viet Nam
[c] Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Viet Nam

ARTICLE INFO

Article history:
Received 1 July 2021
Revised 12 March 2022
Accepted 13 April 2022
Available online 15 May 2022

MSC:
52A30
52B55
68Q25
65D18

Keywords:
Orthogonal convex hulls
Quickhull algorithm
Orthogonal convex sets
Orthogonal polygons
$x-y$ convex hulls
Extreme points

ABSTRACT

The Quickhull algorithm for determining the convex hull of a finite set of points was independently conducted by Eddy in 1977 and Bykat in 1978. Inspired by the idea of this algorithm, we present a new efficient algorithm, for determining the connected orthogonal convex hull of a finite set of points through extreme points of the hull, that still keeps advantages of the Quickhull algorithm. Consequently, our algorithm runs faster than the others (the algorithms introduced by Montuno and Fournier in 1982 and by An, Huyen and Le in 2020). We also show that the expected complexity of the algorithm is $O(n\log n)$, where n is the number of points.

1. Introduction

Orthogonal convexity (rectilinearity, or (x,y)-convexity) is an interesting subject in computational geometry and convex analysis. Starting from the pattern recognition problem (determining whether the shape of an input object has the same type as one of certain groups of objects or not), Unger introduced the concept of orthogonal convex for the first time in 1959 [38]. Then some related names are given in other applications such as rectilinear convex hulls [27], finitely oriented convexity [30], restricted-oriented convexity [16] and convex fuzzy sets [36]. Until the late80′s of the twentieth century, the definition of orthogonal convex set and orthogonal convex hull were clarified and appeared in many applications in different fields, especially in the fields of modern computing and digital imaging [11,34–36,39,40]. In discrete tomography, orthogonal convex hulls are used in the reconstruction of hv-convex discrete sets [8,9] and lattice sets [20]. Some other typical applications of orthogonal convex sets and orthogonal convex hulls include analysis of land-mark data [10], illumination [1], geometric search [33], VLSI circuit layout design [37], shape analysis and classification and many different domains of computer vision

* Corresponding author at: Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Viet Nam.
E-mail addresses: linhnk@ptit.edu.vn (N. Kieu Linh), thanhan@hcmut.edu.vn (P. Thanh An), hoai@hcmut.edu.vn (T. Van Hoai).

https://doi.org/10.1016/j.amc.2022.127183

and pattern recognition [13,19]. In recent years, the problems of image processing, pattern recognition related to orthogonal convex continue to receive more attention from scientists, typically Seo et al. in 2016 [31], Aman et al. in 2020 [4], Alegráa et al. in 2018 and 2021 [2,3], among others. Besides, the geometry optimization problems used in data mining that use orthogonal convexity are recommended. For example, in 2019, Hernn et al. [17] proposed four problems about finding a subset S of a set P consisting of n points in $\mathbb{R}^2$ such that (a) the size of the orthogonal convex hull of S is maximal; (b) the interior of the orthogonal convex hull of S contains no point of P; (c) the interior of the orthogonal convex hull of S contains no point of P and its area is the maximal; (d) the total weight of the points in the set P intersecting the orthogonal convex hull of S is maximal (assuming that each point p in P is weighted $w(p)$ (either negative or positive)).

Although the concept of orthogonal convexity was first mentioned by Unger in 1959 [38], it was not until 1983 that Montuno and Fournier introduced algorithms for computing the (x, y)-convex hulls of a finite set of planar points, an (x, y)-polygon and of a set of (x, y)-polygons under various conditions [24]. The condition under which the (x, y)-convex hull exists is given and an algorithm for testing if the given set of (x, y)-polygons satisfies the condition is presented. Since then, there are several proposed algorithms for solving the orthogonal convex hull problem such as [21,25], and [27]. Unfortunately, the orthogonal convex hull of a set is unique, but it may not be connected. In addition, connected orthogonal convex hull of a ponit set is not unique (see [24] and [27]). Recently, in 2020, An, Huyen and Le prove that if connected orthogonal convex hulls have no semi-isolated points then the connected orthogonal convex hull of a finite point set is unique and the connected orthogonal convex hull can be determined through its extreme points [7]. Their efficient algorithm is modified from the Graham's convex hull algorithm. Recently, Aman et al.'s algorithm [4] is linear for finding the hull for polygons. Nevertheless, it does not work for the finite planar point set.

The Quickhull algorithm [12,15,26] determines the convex hull of a finite point set. The worst case complexity of the algorithm is $O(n^2)$ and its average time is $O(n \log n)$. Furthermore, this algorithm can also be easily designed as a parallel algorithm for finding convex hull of the point set. A survey known algorithms for solving the convex hull of a finite set [23] show that Quickhull algorithm runs faster than some other algorithms and Quickhull algorithm using optimizations proposed by Hoang and Linh [18] has the best performance among the implemented methods and is faster than the current state of the art libraries. In 2019, Linh et al. [22] used successfully the Quickhull algorithm for determining the convex hull of discs.

Recognizing the effectiveness of the Quickhull algorithm, in this paper, we apply the idea of this algorithm and its improved algorithm [18] to propose an algorithm, namely $\mathcal{O}$-QUICKHULL, for determining the extreme points of type j (see Definition 6) of the connected orthogonal convex hull of a set P of n points in $\mathbb{R}^2$ and compare it with the algorithm introduced by An et al. [7] and Montuno and Fournier's algorithm [24]. We also show that the expected complexity of $\mathcal{O}$-QUICKHULL is $O(n \log n)$, where n is the number of points.

The paper consists of several sections. Section 2 presents some concepts of connected orthogonal convexity that will be used in this paper. Section 3 introduces the definition of right orthogonal lines and some other concepts. Section 4 is devoted to $\mathcal{O}$-QUICKHULL algorithm and its expected complexity. Section 5 presents numerical experiments and compares the running time of $\mathcal{O}$-QUICKHULL with some known algorithms.

2. Connected orthogonal convex hulls and their properties

In this paper, we are interested in finding the *connected orthogonal convex hull* of a set P consisting of n points in $\mathbb{R}^2$.

Let be given $p, q, t \in \mathbb{R}^2$, denote $[p, q] := \{(1 - \lambda)p + \lambda q : 0 \le \lambda \le 1\}$, pq the straight line through the points p and q and $\text{dist}(t, pq)$ the Euclidean distance from t to the line pq. We denote by x_p and y_p the x-coordinate and y-coordinate of p respectively. As usual, $\text{dist}(p, q) := \sqrt{(x_p - x_q)^2 + (y_p - y_p)^2}$.

2.1. Connected orthogonal convex hulls

Definition 1 (see [38]). A set $S \subset \mathbb{R}^2$ is *orthogonal convex set* (o. convex set, for short) if and only if for all $c \in \mathbb{R}$ the sets

$$\{S \cap (x, c) : x \in \mathbb{R}\} \text{ and } \{S \cap (c, y) : y \in \mathbb{R}\}$$

are convex.

We call S the *connected orthogonal convex set* (c.o. convex set, for short) if it is both connected and orthogonal convex.

Definition 2 (see [27]). A *connected orthogonal convex hull* (c.o. convex hull, for short) of S is defined to be a minimal c.o. convex set containing S.

Unlike the convex hull of a set, c.o. convex hull of a set may not be unique. For example, Fig. 1 illustrates three c.o. convex hulls of a set of two points a and b.

Let $p = (x_p, y_p), q = (x_q, y_q) \in S \subset \mathbb{R}^2$, the L_1 norm is, by definition,

$$\|p - q\|_1 = |x_p - x_q| + |y_p - y_q|.$$

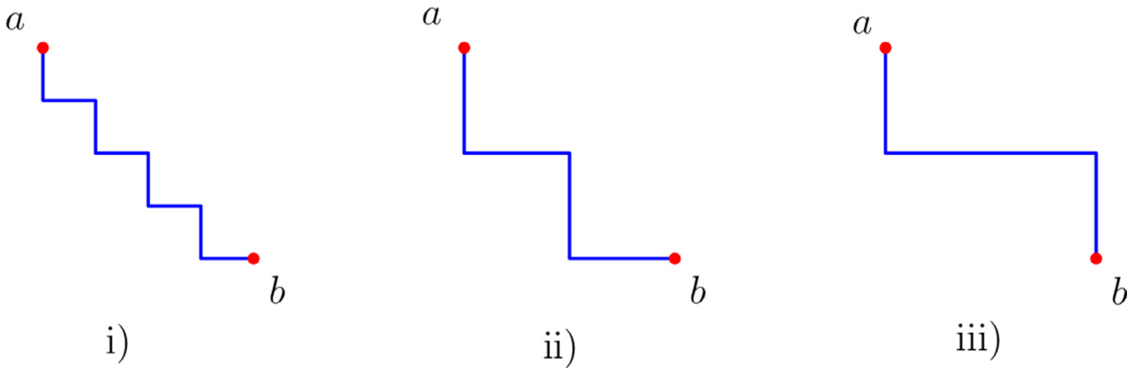


Fig. 1. Three c.o. convex hulls of a set of two points a and b.

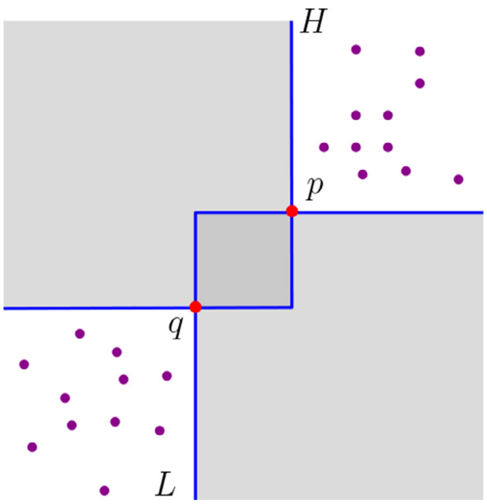


Fig. 2. H and L are two opposite o. supports.

A path that joins p and q with L_1 norm and with the length $\|p-q\|_1$ is call an *orthogonal shortest path* (o. shortest path, for short) from p to q, denoted by $\mathrm{OSP}(p,q)$. An o. shortest path $\mathrm{OSP}(p,q)$ is an *orthogonal increasingly monotone* (o. increasingly monotone, for short) if and only if given any $a,b \in \mathrm{OSP}(p,q)$ one must have $(x_a-x_b)(y_a-y_b) \geq 0$.

We use the definition L_1 norm in Proposition 1 and Lemma 2.

Proposition 1 (see [7]). *A set $S \subset \mathbb{R}^2$ is c.o. convex iff for all $p,q \in S$, there exists an o. increasingly monotone shortest path $\mathrm{OSP}(p,q) \subset S$ joining p and q with L_1 norm.*

A line (half line, line segment, resp.) parallel to either x-axis or y-axis is called a *rectilinear line* (*rectilinear half line, rectilinear line segment*, resp.).

Let $p,q \in \mathbb{R}^2$, an *orthogonal line* (*o. line*, for short) $\mathrm{l}(p,q)$, $(x_p \neq x_q, y_b \neq y_q)$ through p,q is, by definition, the union of two rectilinear half lines with the same starting point, where one rectilinear half line goes through p and the other goes through q. And obviously, if $x_p = x_q$ or $y_p = y_q$ then the o. line $\mathrm{l}(p,q)$ is simply the line through p and q. The common point of two the rectilinear half lines of $\mathrm{l}(p,q)$ is called the *vertex* of $\mathrm{l}(p,q)$. We also denote by $l^v(p,q)$ the o. line $\mathrm{l}(p,q)$ having the vertex v.

Each o. line $\mathrm{l}(p,q)$, $(x_p \neq x_q, y_p \neq y_q)$ divides the space $\mathbb{R}^2$ into two regions, and the quadrant region together with $\mathrm{l}(p,q)$ is called a *orthogonal quadrant* (*o. quadrant*, for short) determined by $\mathrm{l}(p,q)$.

Definition 3 (see [7]). An o. line $\mathrm{l}(p,q)$ is called *orthogonal supporting line* (*o. support*, for short) of a set $S \subset \mathbb{R}^2$ (S may not contain p and q) if its intersection with S is non-empty and all points of $S \setminus (S \cap \mathrm{l}(\mathrm{p},\mathrm{q}))$ either are not on the o. quadrant of $\mathrm{l}(p,q)$ if $x_p \neq x_q, y_p \neq y_q$ or are one open half plane determined by the line $\mathrm{l}(p,q)$ if $x_p = x_q$, or $y_p = y_q$.

We now recall some concepts given in [7]. Two o. supports of a set S that intersect at exactly two distinct points are called *opposite* (see Fig. 2).

Let $\mathcal{F}(S)$ be the set of all c.o. convex hulls of S. If there exist two opposite o. supports of S intersecting at exactly two distinct points u and v, then a point s in the rectangle whose diagonal is $[u,v] (s \neq u, s \neq v)$ is called a *semi-isolated point* of

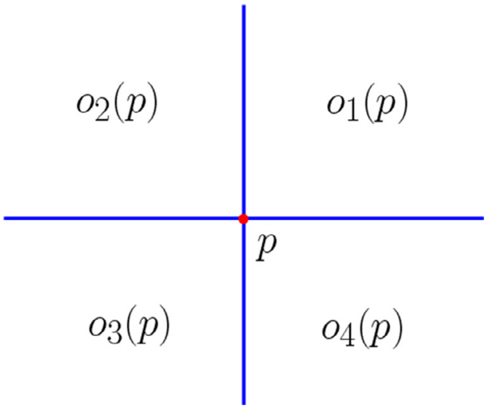


Fig. 3. $o_1(p), o_2(p), o_3(p)$ and $o_4(p)$ are the orthants of the point p.

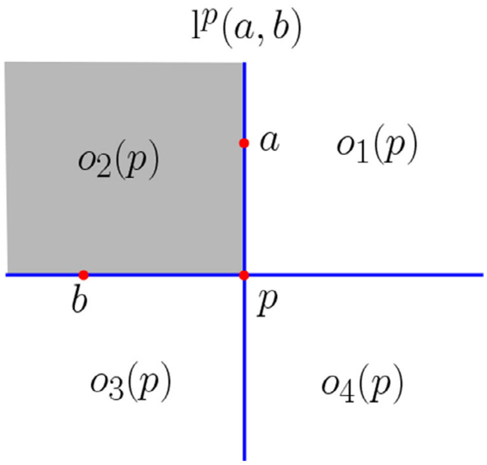


Fig. 4. $o_2(p)$ (shaded area) coincides with the o.quadrant defined by o. line $l^p(a, b)$.

elements of $\mathcal{F}(S)$. From now on, we consider the case that $\mathcal{F}(S)$ has only one element and denote it as COCH(S). According to [7], COCH(S) has no semi-isolated point. This condition ensures that the c.o. convex hull of S is unique [7].

Given a point $p(x_p, y_p)$. The four orthants $o_1(p), o_2(p), o_3(p)$ and $o_4(p)$ are determined by the closed regions

$$o_1(p) := [x_p, +\infty) \times [y_p, +\infty),$$
$$o_2(p) := (-\infty, x_p] \times [y_p, +\infty),$$
$$o_3(p) := (-\infty, x_p] \times (-\infty, y_p],$$
$$o_4(p) := [x_p, +\infty) \times (-\infty, y_p]$$

as the orthants of the point p (see Fig. 3).

Remark 1. It is easy to see that four orthants of a point coincide with the o.quadrants defined by four o. lines of the same point (see Fig. 4).

Let P be a set consisting of n points in $\mathbb{R}^2$. From now, we write c.o. convex hull means c.o. convex hull of a set consisting of n points in $\mathbb{R}^2$. For the remainder of this paper we shall assume that

(A) P consists of a finite number of points in $\mathbb{R}^2$ and the c.o. convex hull of P has no semi-isolated points.

Recall that (A) ensures that the c.o. convex hull of P is unique [7].

Definition 4 (see [14])**.** A point $p \in P$ is called *maximal point* of P if there exists one its orthant does not contain any points of $P \setminus \{p\}$.

Definition 5 (see [7])**.** A point $e \in$ COCH(P) is called an *extreme* of COCH(P) if there exists an orthant of e that does not contain any points of COCH(P) $\setminus \{e\}$. The set of all extreme points of COCH(P) is denoted by ext(COCH(P)).

Definition 6. An point $e \in$ ext(COCH(P)) is of *type* j if $o_j(e)$, $j = 1, 2, 3, 4$, does not contain any point of COCH(P) $\setminus \{e\}$. The set containing all points $e \in$ ext(COCH(P)) of type j is denoted by ext^j(COCH(P)).

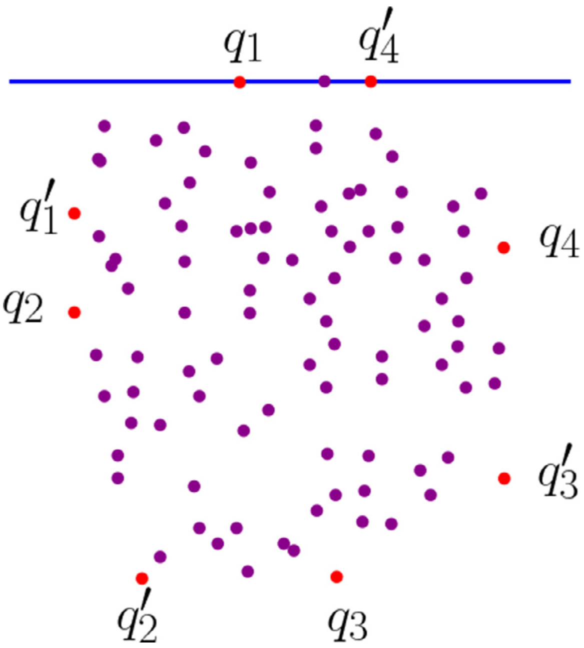


Fig. 5. Eight special points q_1, q_1', q_2, q_2', q_3, q_3', q_4, q_4'.

2.2. Properties of c.o. convex hulls

Definition 7 (see [18])**.** The point with the minimal y-coordinate among the points of P having the maximal x-coordinate is called the rightmost lowest point. Similarly, we define seven other special points of P: highest leftmost, lowest leftmost, lowest rightmost, highest rightmost, leftmost highest, rightmost highest, leftmost lowest points.

In Fig. 5, q_1 is the leftmost highest, q_1' is the highest leftmost, q_2 is the lowest leftmost, q_2' is the leftmost lowest, q_3 is the rightmost lowest, q_3' is the lowest rightmost, q_4 is the highest rightmost, and q_4' is the rightmost highest. It is clear that the eight points in Definition 7 belong to the $\text{ext}(\text{COCH}(P))$.

Remark 2. If $|P| \geq 2$, there exist two distinct points in $\text{ext}(\text{COCH}(P))$. Indeed, we consider the following cases:

i) The statement is obviously true in the case when P consists of exactly 2 disjoint points.
ii) If P has more than two distinct points and the points of P belong to a straight line, then the two ending points are the two distinct extreme points of $\text{COCH}(P)$.
iii) If P has more than two distinct points and the points of P are not collinear, then two distinct extreme points of $\text{COCH}(P)$ can be selected in one of the following ways:
   - the highest leftmost and the lowest rightmost,
   - the leftmost highest and the rightmost lowest,
   - the rightmost highest and the leftmost lowest,
   - the highest rightmost and the lowest leftmost.

Lemma 1 (see [7])**.** *We have*

i) $\text{ext}(\text{COCH}(P)) \subseteq P$.
ii) $P \subseteq \text{COCH}(P)$.
iii) *If* $P_1, P_2 \subseteq \mathbb{R}^2$ *are two finite point sets and* $P_1 \subseteq P_2$*, then* $\text{COCH}(P_1) \subseteq \text{COCH}(P_2)$.

Given a set of points in $\mathbb{R}^2$, the minimum rectilinear rectangle of this set is a minimum rectangle having edges parallel to x or y axises that contains the set.

Lemma 2. *Let* $P := \{p_1, \ldots, p_m\} \subset \mathbb{R}^2$*. Every c.o. convex hull of P is included in its minimum rectilinear rectangle.*

Proof. Let E be a c.o. convex hull of P, R be the minimum rectilinear rectangle bounded of P. Assume, on contrary, $E \not\subseteq R$. Let $F = R \cap E \subsetneq E$. For all $a, b \in F$, $a, b \in E$. By orthogonal convexity of E, Proposition 1 yields that there exists an o. increasingly monotone shortest path γ joining a and b such that $\gamma \subseteq E$. Since $a, b \in R$, γ is a o. increasingly monotone shortest path in $R \cap E = F$, F is connected. We are in position to prove that F is c.o. convex. Let h be an arbitrary horizontal line intersecting F (the case of h being a vertical line is similar). Then $S = h \cap E$ convex. Since $h \cap F = h \cap (R \cap E) = (h \cap E) \cap R = S \cap R$, we

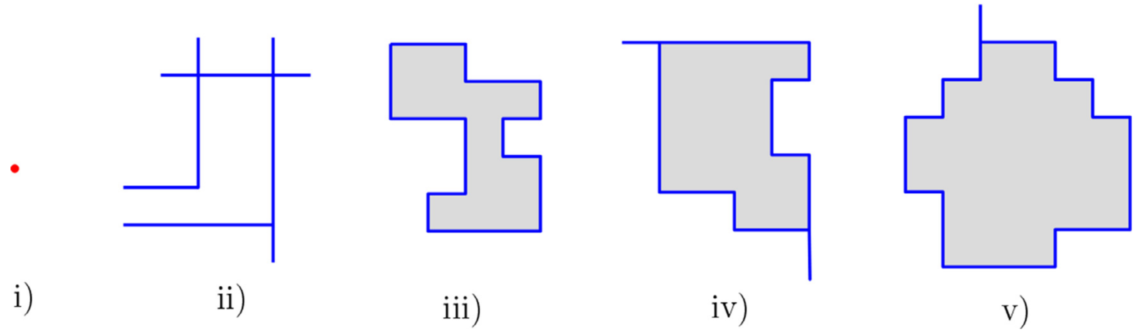


Fig. 6. (i)-(iv) Orthogonal polygons; (v) Orthogonal convex polygon.

conclude that $h \cap F$ is convex. Therefore, F is c.o. convex and $F \subsetneq E$. This contradicts the fact that E is a smallest c.o. convex hull of P. Hence, $E \subseteq R$. $\square$

Note that Proposition 1 and Lemma 2 imply the following Proposition 2 and Proposition 2 is used in Section 5.2.

Proposition 2. *Let $P := \{p_1, \ldots, p_m\} \subset \mathbb{R}^2$. Then, every c.o. convex hull of P is compact.*

The proof of this proposition is given in the Appendix. The compactness of the c.o. convex hull of P is also shown in the Lemma 3 below.

An *orthogonal polygon* (*$(x, y)-$polygon*) is defined as a point, or connected rectilinear line segments, or a simple polygon whose edges are rectilinear, or a connected union of connected rectilinear line segments and/or a simple polygon whose edges are rectilinear (see [24]). An orthogonal polygon is called an *orthogonal convex polygon* (o. convex polygon, for short) if it is o. convex. Some examples of orthogonal polygons and orthogonal convex polygons. are illustrated in Fig. 6.

Lemma 3 (see [7]). $\mathrm{COCH}(P)$ *is an o. convex polygon whose boundary is the union of a finite set of o. supports that pass through two distinct points in* $\mathrm{ext}(\mathrm{COCH(P)})$.

Lemma 4. *If $p \in \mathrm{COCH}(P)$ then each orthant of p contains at least an extreme point of* $\mathrm{COCH}(P)$.

The proof of Lemma 4 is given in the Appendix. We denote the set of maximal points of P by $\mathcal{M}(P)$.

Lemma 5. *Let P satisfy the assumption (A). Then a point of $\mathcal{M}(P)$ is a point in $\mathrm{ext}(\mathrm{COCH}(P))$ and vice versa. Consequently,*

$$\mathcal{M}(P) = \mathrm{ext}(\mathrm{COCH}(P)).$$

The proof of Lemma 5 is given in the Appendix. The following lemma is needed to prove the complexity of the algorithm in Section 4.

Lemma 6 (see [14]). *Let P be the set of n points chosen according to any probability distribution Δ. Then the expected number of maximal points of P is $O(\log n)$.*

3. Right o. lines and some related concepts

In this content we present definition a *right o. line* and some other properties necessary to serve the following section.
Given an ordered triple of points (a, b, c) in $\mathbb{R}^2$, let

$$\mathrm{orient}(a, b, c) = \begin{vmatrix} 1 & x_a & y_a \\ 1 & x_b & y_b \\ 1 & x_c & y_c \end{vmatrix}. \tag{1}$$

Definition 8 (see [26]). We say that

(i) The ordered triple (a, b, c) has *positive orientation* (*negative orientation, zero orientation*, resp.) if $\mathrm{orient}(a, b, c) > 0$ ($\mathrm{orient}(a, b, c) < 0$, $\mathrm{orient}(a, b, c) = 0$, resp.).

(ii) The point c is called *on the left of* (*on the right of, on*, resp.) the directed line ab if $\mathrm{orient}(a, b, c) > 0$ ($\mathrm{orient}(a, b, c) < 0$, $\mathrm{orient}(a, b, c) = 0$, resp.).

Definition 9. Let $\mathrm{l}^v(a, b)$ be the o. line through two points a and b ($x_a \neq x_b, y_a \neq y_b$) with its vertex v. If b is on the right of av then we call $l^v(a, b)$ the *right o. line* from a to b and denoted it by $\mathcal{L}_R^v(a, b)$ (Fig. 7).

Definition 10. Let $\mathcal{L}_R^v(a, b)$ be a right o. line from a to b with its vertex v. A point p is called *is on the right of $\mathcal{L}^v(a, b)$* if p is on the right of both av and vb. A point p is called *is on the left of $\mathcal{L}^v(a, b)$* if p is either on the left of av or on the left of vb (Fig. 7(i)).

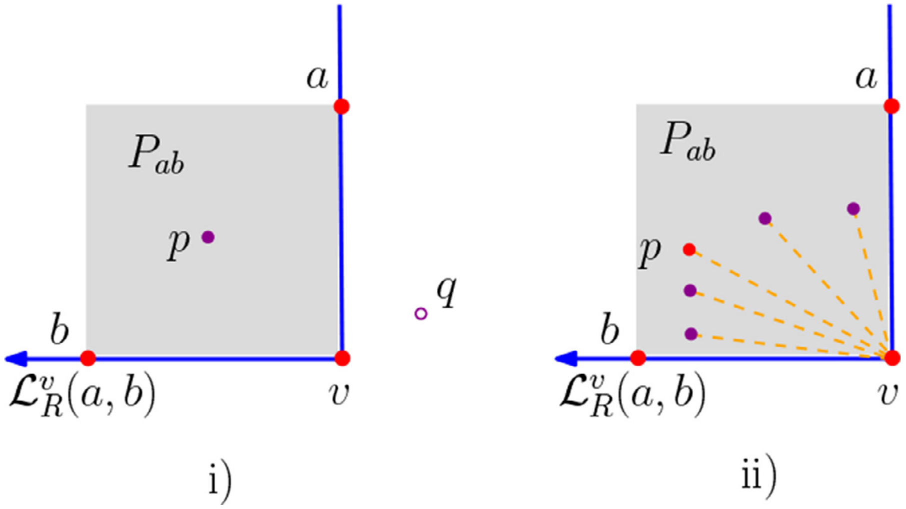


Fig. 7. (i) p is on the right of $\mathcal{L}^v(a,b)$, q is on the left of $\mathcal{L}^v(a,b)$; (ii) p is the farthest point to $\mathcal{L}^v(a,b)$.

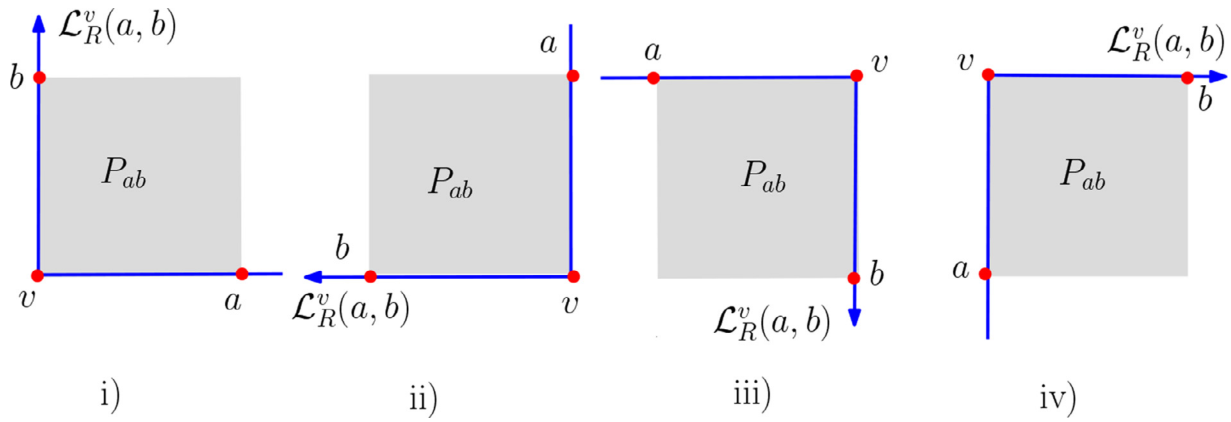


Fig. 8. The four cases of a, b: (i) Case 1; (ii) Case 2; (iii) Case 3; (iv) Case 4.

We denote by P_{ab} the set containing all points of P being on the right of $\mathcal{L}_R^v(a,b)$.

Definition 11. Let $\mathcal{L}_R^v(a,b)$ be a right o. line from a to b with its vertex v and $p \in P_{ab}$. We call the length of $[p,v]$ the *orthogonal distance* from p to $\mathcal{L}_R^v(a,b)$, denoted by $\mathrm{Odist}(p, \mathcal{L}_R^v(a,b))$. The point p is called the *farthest point* of P_{ab} to $\mathcal{L}_R^v(a,b)$ if p satisfies

$$\mathrm{Odist}(p, \mathcal{L}_R^v(a,b)) = \max_{q \in P_{ab}}\{\mathrm{Odist}(q, \mathcal{L}_R^v(a,b))\}.$$

Note from Definition 6 that an extreme point e of $\mathrm{COCH}(P)$ is of type j, $j = 1,2,3,4$, if the orthant $o_j(e)$ does not contain any points of $\mathrm{COCH}(P) \setminus \{e\}$ and the set containing all the extreme point of $\mathrm{COCH}(P)$ of type j is denoted by $\mathrm{ext}^j(\mathrm{COCH}(P))$.

Lemma 7. *Let a, b ($x_a \neq x_b, y_a \neq y_b$) be any two distinct extreme points of COCH(P). Then*

$$\mathrm{ext}_{ab}^j(\mathrm{COCH}(P)) \subseteq \mathrm{ext}(\mathrm{COCH}(P_{ab})),$$

where $\mathrm{ext}_{ab}^j(\mathrm{COCH}(P))$ the set of all the extreme points of type j of $\mathrm{COCH}(P)$ in P_{ab}.

Proof. Let $e \in \mathrm{ext}_{ab}^j(\mathrm{COCH}(P))$. According to Definition 6, the orthant $o_j(e)$ does not contain any points of $\mathrm{COCH}(P) \setminus \{e\}$. Because $P_{ab} \subseteq P$ and Lemma 1(iii), we get that $\mathrm{COCH}(P_{ab}) \subseteq \mathrm{COCH}(P)$ and therefore $o_j(e)$ does not contain any points of $\mathrm{COCH}(P_{ab}) \setminus \{e\}$. It follows that $e \in \mathrm{ext}(\mathrm{COCH}(P_{ab}))$, i.e, $\mathrm{ext}_{ab}^j(\mathrm{COCH}(P)) \subseteq \mathrm{ext}(\mathrm{COCH}(P_{ab}))$. □

Let a, b ($x_a \neq x_b, y_a \neq y_b$) be any two distinct extreme points of $\mathrm{COCH}(P)$. There are four cases of two points a and b as follows (see Fig. 8).

- Case 1: $x_a > x_b$ and $y_a < y_b$;

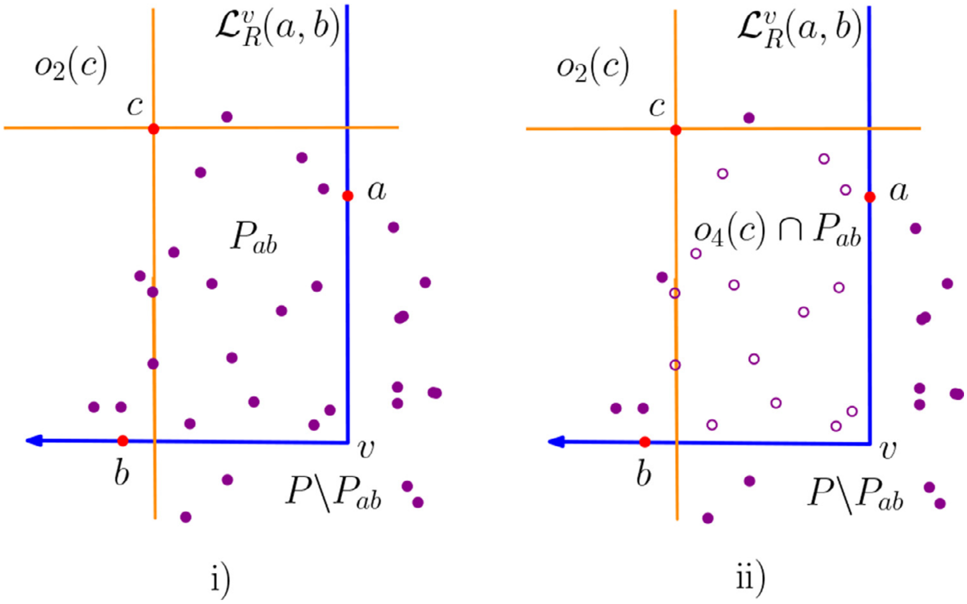


Fig. 9. In the Case 2 of a, b (i) the farthest point c to $\mathcal{L}^v(a,b)$ is a point of $\mathrm{ext}^2(\mathrm{COCH}(P))$ and (ii) all the points in the set $o_4(c) \cap (P_{ab} \setminus \{c\})$ are not in $\mathrm{ext}^j_{ab}(\mathrm{COCH}(P))$.

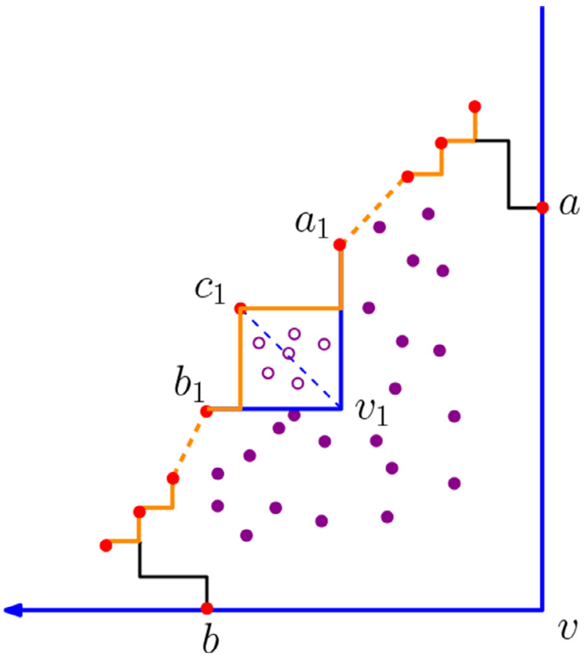


Fig. 10. Three extreme points a_1, c_1, b_1 of type 2.

- Case 2: $x_a > x_b$ and $y_a > y_b$;
- Case 3: $x_a < x_b$ and $y_a > y_b$;
- Case 4: $x_a < x_b$ and $y_a < y_b$.

The following proposition is needed to prove the correctness of the algorithm in the next section.

Proposition 3. *Let a, b ($x_a \neq x_b, y_a \neq y_b$) be any two distinct extreme points of* $\mathrm{COCH}(P)$ *which are in Case j $j = 1, 2, 3, 4$ and c be a farthest point of P_{ab} from the right o. line $\mathcal{L}_R^v(a,b)$. Then*

i) $c \in \mathrm{ext}^j_{ab}(\mathrm{COCH}(P))$.

Algorithm 1 $\mathcal{O}$-QUICKHULL algorithm.

Function $\mathcal{O}$-Quickhull(a, b, P_{ab})
(a, b are any two distinct extreme points of COCH(P) and suppose that they are in the Case j, $j = 1, 2, 3, 4$; P_{ab} is the set of all points on the right of the right o.~line $\mathcal{L}_R^v(a, b)$ with its vertex v from a to b)

1. **if** $P_{ab} = \emptyset$ then **return** ()
2. **else**
   (a) $c \leftarrow$ the farthest point from $\mathcal{L}_R^v(a, b)$.
   (b) **if** a, c are in the Case j **then** $P_{ac} \leftarrow$ the set of points on the right of the right o.~line $\mathcal{L}_R^{v_1}(a, c)$ from a to c with its vertex v_1.
   (c) **if** c, b are in the Case j **then** $P_{cb} \leftarrow$ the set of points on the right of the right o.~line $\mathcal{L}_R^{v_2}(c, b)$ from c to b with its vertex v_2.
   (d) **return** $\mathcal{O}$-Quickhull$(a, c, P_{ac}) \cup \{c\} \cup \mathcal{O}$-Quickhull$(c, b, P_{cb})$.

ii) $o_v(c) \cap (P_{ab} \setminus \{c\}) \cap \text{ext}^j(\text{COCH}(P)) = \emptyset$, *where* $o_v(c)$ *is an orthant of* c *and contains* v. *Consequently, any point in* $o_v(c) \cap (P_{ab} \setminus \{c\})$ *is not in* $\text{ext}_{ab}^j(\text{COCH}(P))$.

Proof. Consider the Case 2 ($j = 2$) of two points a, b (the other cases, $j = 1, 3, 4$, are similar). (i) We claim that

$$o_2(c) \text{ does not contain any point of } P_{ab} \setminus \{c\}. \tag{2}$$

Assume the contrary that there is a point $t \in o_2(c) \cap (P_{ab} \setminus \{c\})$. Then we have $x_t \le x_c$, $y_t \ge y_c$, and $t \ne c$. We get dist$(t, s) >$ dist(c, s) and therefore c is not a farthest point of P_{ab} from the right o. line $\mathcal{L}^s(a, b)$, a contradiction. Thus (2) holds true.

On the other side, if a point $t \in P \backslash P_{ab}$ then $x_t \ge x_a$ or $y_t \le y_b$. It follows that $x_t > x_c$ or $y_t < y_c$, i.e., $t \notin o_2(c)$. Therefore,

$$o_2(c) \text{ does not contain any point of } P \backslash P_{ab}. \tag{3}$$

From (2) and (3) we deduce that

$$o_2(c) \text{ does not contain any point of } P \setminus \{c\}, \tag{4}$$

i.e, $c \in \mathcal{M}(P)$. It follows from the Lemma 5 and (4) that $c \in \text{ext}^2(\text{COCH}(P))$. ii) In the Case 2 of two point a and b, we get that $o_v(c) \equiv o_4(c)$. Take $d \in o_4(c) \cap (P_{ab} \setminus \{c\})$. We get that $x_c \le x_d$ and $y_c \ge y_d$ and therefore $c \in o_2(d)$. It follows from Definition 6 that $d \notin \text{ext}_{ab}^2(\text{COCH}(P))$. □

Remark 3. Among the points of the set P_{ab}, there can be more than one farthest point from the right o. line $\mathcal{L}_R^v(a, b)$. However, all of them are the extreme points of COCH(P) (according to Proposition 3(i)).

4. Algorithm based on Quickhull for finding the c.o. convex hulls

4.1. $\mathcal{O}$-QUICKHULL algorithm

Consider the following four cases:

- the leftmost highest concides with the highest leftmost,
- the lowest leftmost concides with the leftmost lowest,
- the rightmost lowest concides with the lowest rightmost,
- the highest rightmost concides with the rightmost highest.

If all four of these conditions hold, then COCH(P) is a rectangle formed by these points. Therefore, we assume from now on that at least one of these conditions does not hold.

Inspired by the idea of the Quickhull algorithm [12,15,26], we now present a new efficient algorithm, namely $\mathcal{O}$-QUICKHULL, for finding the c.o. convex hull COCH(P) of P under the assumption (A). The first step of the $\mathcal{O}$-QUICKHULL is to find two distinct extreme points, say a and b, of COCH(P) (this is always guaranteed according to Remark 2). Let $\mathcal{L}_R^v(a, b)$ be the o. line with its vertex v from a to b. Note that, P_{ab} is the set containing all points on the right of $\mathcal{L}_R^v(a, b)$. Then, from P_{ab} find the point c that is the farthest point from the right o. line $\mathcal{L}_R^v(a, b)$. Put the point c in the ext(COCH(P)). Let $\mathcal{L}_R^{v_1}(a, c)$ ($\mathcal{L}_R^{v_2}(c, b)$, resp.) be the right o. line with its vertex v_1 (v_2, resp.) from a to c (from c to b). Proposition 3(ii) allows us not to consider points $t \in o_v(c) \cap (P_{ab} \setminus \{c\})$. Therefore, to find the next extreme points of COCH(P), we replace the right o. line $\mathcal{L}_R^v(a, b)$ by $\mathcal{L}_R^{v_1}(a, c)$ and $\mathcal{L}_R^{v_2}(c, b)$, and recursively continue the algorithm.

$\mathcal{O}$-QUICKHULL illustrates the function $\mathcal{O}$-Quickhull(a, b, P_{ab}), where $a \ne b$ and $a, b \in \text{ext}(\text{COCH}(P))$ and P_{ab} is the set of all points on the right of $\mathcal{L}_R^v(a, b)$. If two points a, b are in Case j ($j = 1, 2, 3, 4$) then the output of $\mathcal{O}$-QUICKHULL contains all the extreme points of type j of COCH(P) in P_{ab}. We use "∪" to represent list concatenation. The final o. convex hull is found when we choose pairs of extreme points to apply $\mathcal{O}$-QUICKHULL (see Section 5.2).

Remark 4. In $\mathcal{O}$-Quickhull, after finding the extreme point c of type j, to find the next extreme points of type j, we just need to consider the points in the set P_{ac} and P_{cb}, that is, the points in the set $P_{ab} \setminus ((P_{ac} \cup P_{cb}) \setminus \{c\})$ are not considered anymore because these points cannot be the extreme points of type j of the COCH(P) in P_{ab})(see Proposition 3(ii)).

4.2. the correctness and the complexity of $\mathcal{O}$-Quickhull

We now present the correctness of $\mathcal{O}$-Quickhull in the following

Theorem 1. *If two points a, b are in Case j $(j = 1, 2, 3, 4)$ then the output of $\mathcal{O}$-Quickhull contains all the extreme points of type j of COCH(P) in P_{ab}.*

Proof. If $|P_{ab}| = 0$, then the output of $\mathcal{O}$-Quickhull has no points, which is the satisfied conclusion.

If $|P_{ab}| \geq 1$, the function $\mathcal{O}$-Quickhull(a, b, P_{ab}) of $\mathcal{O}$-Quickhull chooses the farthest point c from $\mathcal{L}^{v}(a, b)$ and therefore $c \in \text{ext}^{j}(\text{COCH}(P))$ (Proposition 3(i)).

We prove on the contrary that if $c_1 \in P_{ab}, c_1 \notin \{a, b\}$, $c_1 \in \text{ext}^{j}(\text{COCH}(P))$ then c_1 is also generated by some step in $\mathcal{O}$-Quickhull, that is, there exist $a_1 \neq b_1$, a_1, b_1 in Case j and $a_1, b_1 \in \text{ext}^{j}(\text{COCH}(P))$ such that c_1 is the farthest point from the right o. line $\mathcal{L}_R^{v_1}(a_1, b_1)$. Indeed, a_1 (b_1, resp.) can be chosen as the extreme point with the smallest x-coordinate (largest y-coordinate, resp.) of the extreme points with the same type j but its x-coordinate (y-coordinate, resp.) is greater (less, resp.) than that of c_1. Consider Case $j = 2$ (the other cases are similar). Taking a point $t \in P_{a_1 b_1}$, we claim that

$$t \text{ is in the rectangle with the diagonal } [c_1, v_1]. \tag{5}$$

Assume the contrary that t is outside the rectangle with the diagonal $[c_1, v_1]$, i.e., t in $o_2(c_1)$ or $P_{a_1 c_1}$ or $P_{c_1 b_1}$. If $t \in o_2(c_1)$ then this contradicts the assumption that $c_1 \in \text{ext}^{2}(\text{COCH}(P))$. If $t \in P_{a_1 c_1}$ (or $t \in P_{c_1 b_1}$, resp.), i.e., $P_{a_1 c_1} \neq \emptyset$ (or $P_{c_1 b_1} \neq \emptyset$, resp.), then there exists a farthest point u of $P_{a_1 c_1}$ (or $P_{c_1 b_1}$, resp.) from the right o. line from a_1 to c_1 (or from c_1 to b_1, resp.). According to Proposition 3(i), $u \in \text{ext}^{2}_{a_1 c_1}(\text{COCH}(P))$ (or $u \in \text{ext}^{2}_{c_1 b_1}(\text{COCH}(P))$, resp.). It follows that $x_{c_1} < x_u < x_{a_1}$ (or $y_{b_1} < y_u < y_{c_1}$), resp.). This contradicts the choice of a_1 and b_1. Thus (5) holds true. It follows that $\text{dist}(t, v_1) \leq \text{dist}(c_1, v_1)$ and therefore c_1 is the farthest point from the right o. line $\mathcal{L}_R^{v_1}(a_1, b_1)$. □

Theorem 2. *Suppose that the set P_{ab} consists n points. The worst case complexity of the $\mathcal{O}$-Quickhull is $O(n^2)$ and its expected complexity is $O(n \log n)$.*

Proof. Suppose that the output of $\mathcal{O}$-Quickhull has m extreme points of type j of COCH(P) in P_{ab}. The algorithm calls $\mathcal{O}$-Quickhull functions $(m + 1)$ times. In which, each of the first m functions finds exactly one extreme point of type j of COCH(P) and need $O(n)$ time complexity. The last call to the $\mathcal{O}$-Quickhull function works with an empty set. So the time complexity of $\mathcal{O}$-Quickhull is $O(mn)$. In the worst case, when $m = n$, we have the worst time complexity of $O(n^2)$. According to Lemmas 5, 6, and 7, the expected number of extreme points of type j of COCH(P) in P_{ab} is $O(\log n)$, i.e., $m = O(\log n)$. Therefore, the expected complexity of $\mathcal{O}$-Quickhull is $O(n \log n)$. □

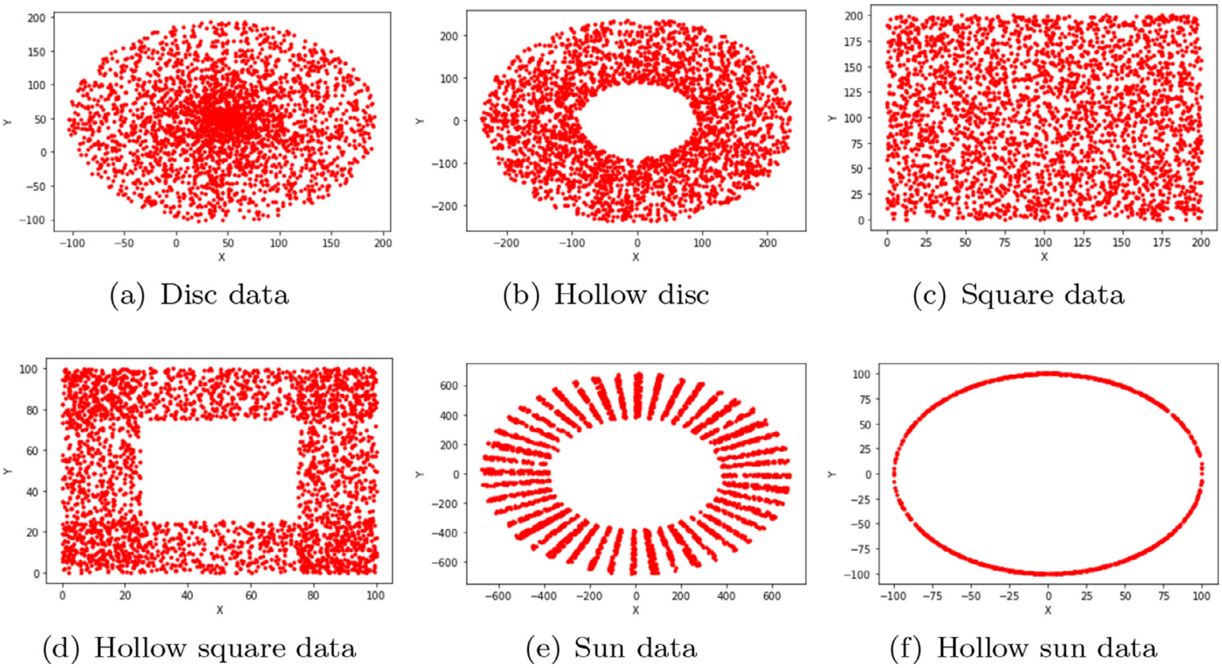
5. Implementation

In this section, we present the selection of pairs of distinct extreme points a and b to apply $\mathcal{O}$-Quickhull to find the final c.o. convex hull. Besides, we are going to compare the running times of $\mathcal{O}$-Quickhull to $\mathcal{O}$-Graham introduced by An, Huyen and Le in [7] and an other algorithm proposed by Montuno and Fournier in [24].

5.1. The test sets

To test the algorithms we create six following data types.

- Disc data: We create random real points inside a disc (see Fig. 11 (a)).
- Hollow disc data: First we generate two discs (or ellipses) with the same center and different radius. The data points will be placed randomly outside the smaller disc (or ellipse) and inside the bigger one (see Fig. 11 (b)).
- Square data: The data points are random real points in inside a square (see Fig. 11 (c)).
- Hollow square data: The way to create sets of points is similar to Hollow disc data but replace concentric discs (or ellipses) by concentric squares (see Fig. 11 (d)).
- Hollow sun data: The points are randomly generated according to the central angle of a disc and interspersed equal angles with no points. In addition, these points are also created outside the concentric disc with the one above (see Fig. 11 (e)).
- Circle data: We create random real points on a circle (see Fig. 11 (f)).



(a) Disc data (b) Hollow disc (c) Square data

(d) Hollow square data (e) Sun data (f) Hollow sun data

Fig. 11. Six data types.

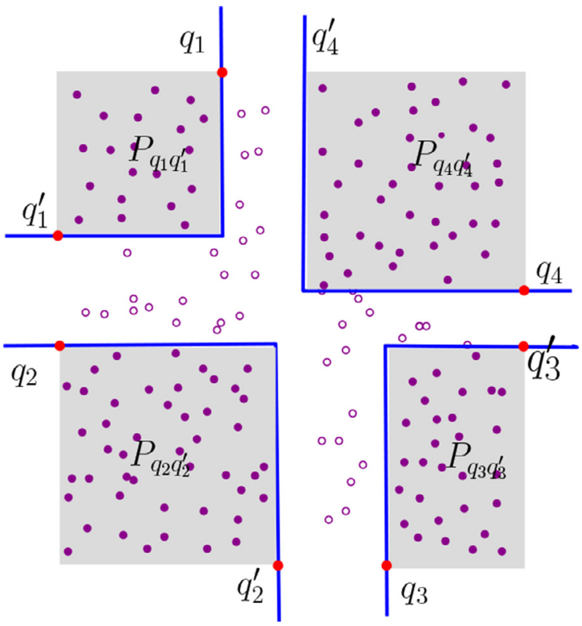


Fig. 12. Four sets $P_{q_iq'_i}$, $i = 1, 2, 3, 4$.

5.2. Numerical results

In this subsection we present the selection of pairs of distinct extreme points a and b to apply $\mathcal{O}$-Quickhull to find the final c.o. convex hull.

It is known that the points lying inside or on the edges of the o. polygon formed by eight extreme points q_1, q'_1, q_2, q'_2, q_3, q'_3, q_4 and q'_4 (except the eight these points) are not in ext(COCH(P)). So we can delete these points and consider it as a preprocessing step (see Fig. 12). This preprocessing step is applied for all methods tested.

Due to the compactness of COCH(P), we can determine its boundary according to Lemma 3. Thus COCH(P) is an o. convex (x, y)-polygon whose boundary is union of the rectilinear line segments $[q_i, q'_i]$, $i = 1, 2, 3, 4$ and staircase paths $\mathcal{P}_{q_iq'_i}$

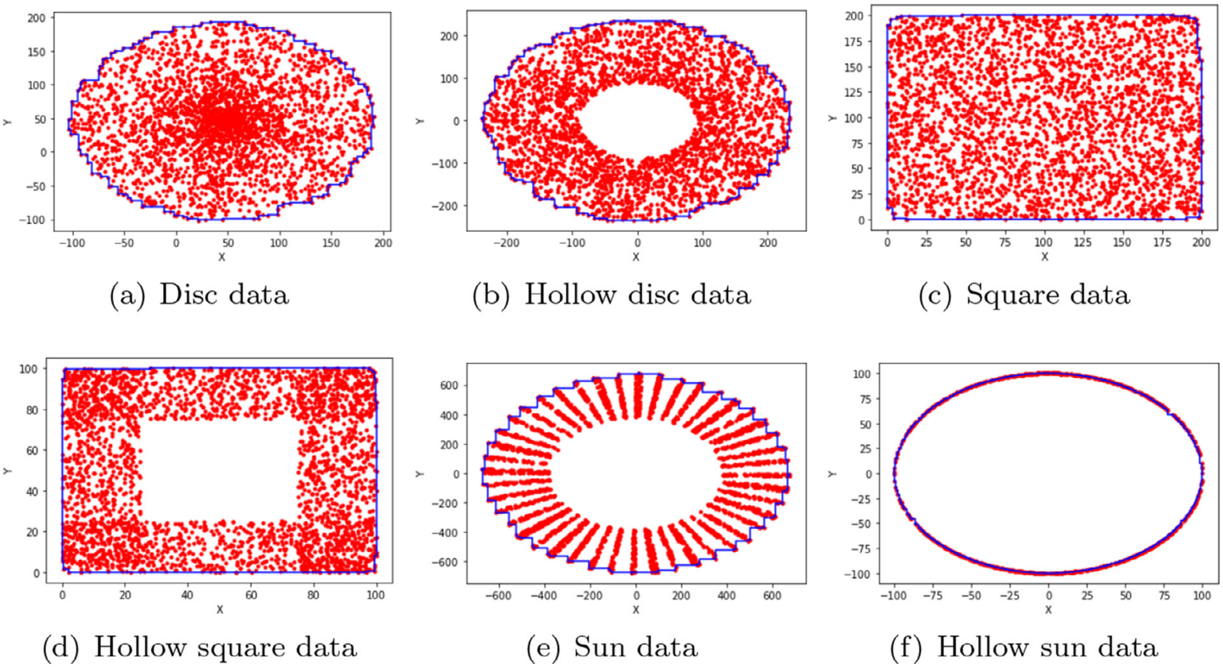


Fig. 13. The c.o. convex hull of six data sets.

Table 1
The actual running times (in seconds) of the algorithms for square data.

Input	$\mathcal{O}$-Graham	Montuno & Fournier's algorithm	$\mathcal{O}$-Quickhull
1000	0.0024	0.0015	0.0012
5000	0.0054	0.0075	0.0017
10,000	0.0034	0.0026	0.0018
50,000	0.0037	0.0031	0.0025
100,000	0.0039	0.0032	0.0032
500,000	0.0098	0.0097	0.0082
1,000,000	0.0193	0.0188	0.0187
5,000,000	0.2064	0.1837	0.1770
10,000,000	0.4502	0.4285	0.3712
20,000,000	0.8702	0.8002	0.7500
30,000,000	1.3246	1.2046	1.1102
40,000,000	1.7892	1.6892	1.5429
50,000,000	2.2785	2.1752	1.8958

(formed by the extreme points with the same type) joining q_i and q_i', $i = 1, 2, 3, 4$, respectively. We apply $\mathcal{O}$-Quickhull for set $P_{q_1q_1'}$ if $q_1 \neq q_1'$, set $P_{q_2q_2'}$ if $q_2 \neq q_2'$, set $P_{q_3q_3'}$ if $q_3 \neq q_3'$, set $P_{q_4q_4'}$ if $q_4 \neq q_4'$. Therefore, the final c.o. convex hull COCH(P) is

$$\{q_1\} \cup \mathcal{O}\text{-Quickhull}(q_1, q_1', P_{q_1q_1'}) \cup \{q_1', q_2\} \cup \mathcal{O}\text{-Quickhull}(q_2, q_2', P_{q_2q_2'})$$

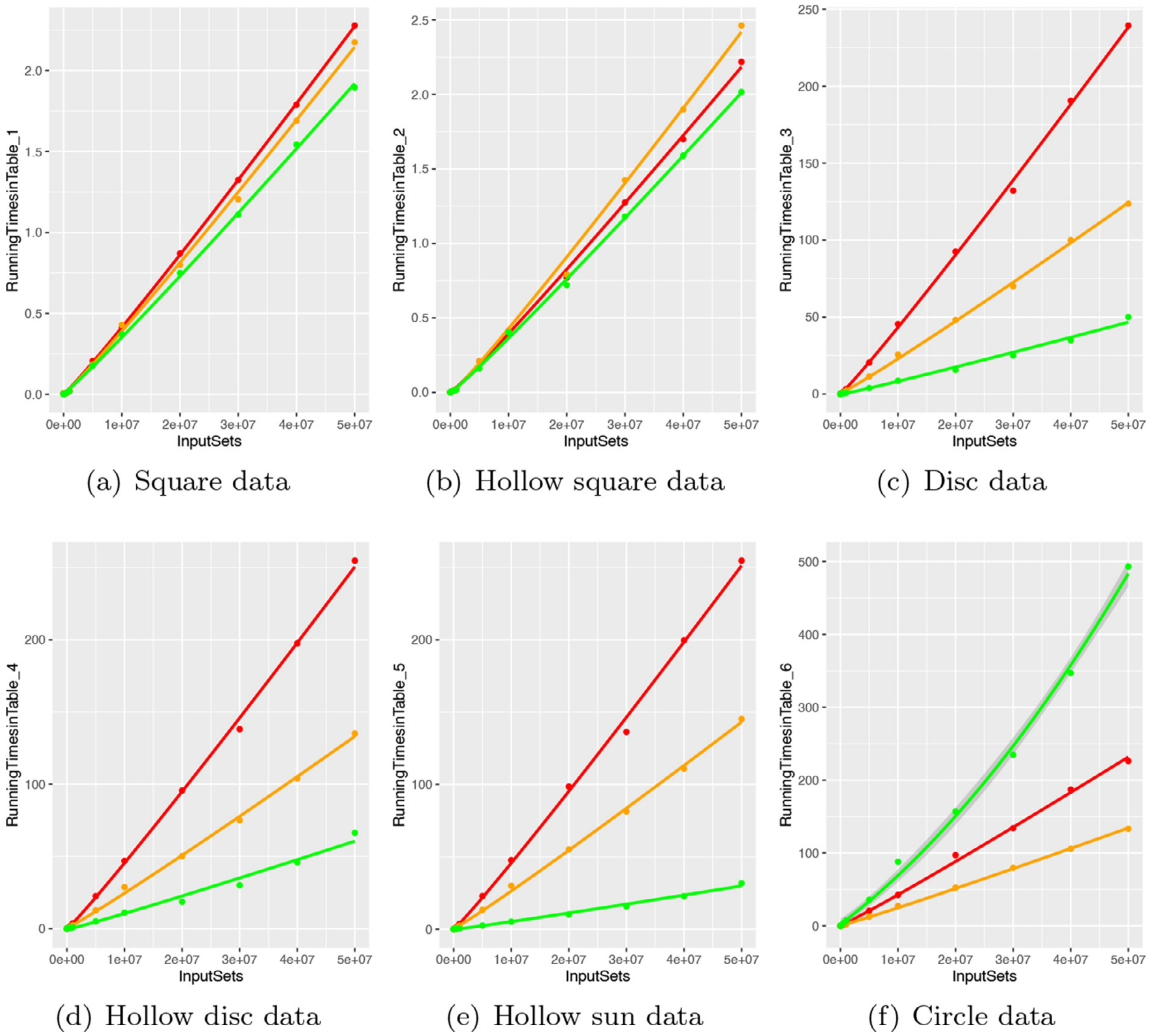
$$\cup \{q_2', q_3\} \cup \mathcal{O}\text{-Quickhull}(q_3, q_3', P_{q_3q_3'}) \cup \{q_3', q_4\} \cup \mathcal{O}\text{-Quickhull}(q_4, q_4', P_{q_4q_4'}).$$

If no case occurs (i.e., $q_1 = q_1'$, $q_2 = q_2'$, $q_3 = q_3'$, $q_4 = q_4'$), the rectangle $q_1q_2q_3q_4$ is the c.o. convex hull to look for.

The algorithms are implemented in Python and run on PC Intel(R) Xeon(R) CPU E5-4627 v4 @ 2.60GHz with 64GB RAM. All codes are given at the link https://github.com/linhnk2109/O-Convexhull.

Fig. 13 illustrates the results of finding c.o. convex hull of the sets of points corresponding to the data sets. The pre-processing time is included in all timing results. Tables 1–5 list the running times (in seconds) of the three algorithms: $\mathcal{O}$-Quickhull, $\mathcal{O}$-Graham introduced by An, Huyen and Le in [7] ($\mathcal{O}$-Graham, in short) and the algorithm proposed by Montuno and Fournier in [24] (Montuno and Fournier's algorithm, in short).

The geometric mean and regression are used to analyze the data in Tables 1–5. In general, both analyzes show that $\mathcal{O}$-Quickhull runs faster than the other two algorithms. However, in Table 5, we test for datasets forming, we test for datasets created on circles, whose all points are vertices of c.o. convex hull. Therefore, this data type will make the $\mathcal{O}$-Quickhull algorithm the worst complexity (According to Theorem 2, $\mathcal{O}$-Quickhull algorithm in this case has a computational complexity



(a) Square data (b) Hollow square data (c) Disc data

(d) Hollow disc data (e) Hollow sun data (f) Circle data

Fig. 14. The best fit curves.

Table 2
The actual running times (in seconds) of the algorithms for hollow square data.

Input	$\mathcal{O}$-Graham	Montuno & Fournier's algorithm	$\mathcal{O}$-QUICKHULL
1,000	0.00143	0.00142	0.00121
5,000	0.00204	0.00222	0.00155
10,000	0.00132	0.00111	0.00107
50,000	0.00188	0.00186	0.00133
100,000	0.00255	0.00266	0.00155
500,000	0.01020	0.01037	0.01001
1,000,000	0.02107	0.01936	0.01582
5,000,000	0.20826	0.20695	0.16207
10,000,000	0.41317	0.40864	0.40374
20,000,000	0.7702	0.8915	0.7202
30,000,000	1.2746	1.4246	1.1805
40,000,000	1.6989	1.8989	1.5889
50,000,000	2.21878	2.46234	2.01633

Table 3
The actual running times (in seconds) of the algorithms for disc data.

Input	$\mathcal{O}$-Graham	Montuno & Fournier's algorithm	$\mathcal{O}$-Quickhull
1000	0.0033	0.0022	0.0030
5000	0.0158	0.0090	0.0071
10,000	0.0307	0.0208	0.0103
50,000	0.1430	0.1001	0.0356
100,000	0.2247	0.1672	0.0705
500,000	1.3982	0.9295	0.3693
1,000,000	3.2462	2.0679	0.6839
5,000,000	20.3982	11.2869	3.8614
10,000,000	45.4468	25.5363	8.5631
20,000,000	92.5764	48.0936	15.6375
30,000,000	132.0943	70.0291	24.9875
40,000,000	190.5429	99.9872	34.9083
50,000,000	239.4378	123.6434	50.0541

Table 4
The actual running times (in seconds) of the algorithms for hollow disc data.

Input	$\mathcal{O}$-Graham	Montuno & Fournier's algorithm	$\mathcal{O}$-Quickhull
1,000	0.0027	0.0022	0.0017
5000	0.0134	0.0104	0.0060
10,000	0.0256	0.0211	0.0114
50,000	0.1205	0.1063	0.0456
100,000	0.2358	0.1750	0.0890
500,000	1.5202	1.0174	0.4439
1,000,000	3.6673	2.3335	0.8788
5,000,000	23.4799	12.6681	5.0595
10,000,000	45.8641	28.6821	11.0427
20,000,000	95.2714	50.0906	18.5175
30,000,000	138.3943	75.1191	29.2975
40,000,000	197.1429	103.0272	45.8383
50,000,000	254.8013	135.0761	66.2740

Table 5
The actual running times (in seconds) of the algorithms for hollow sun data.

Input	$\mathcal{O}$-Graham	Montuno & Fournier's algorithm	$\mathcal{O}$-Quickhull
1,000	0.00328	0.00220	0.00354
5,000	0.01614	0.01037	0.00662
10,000	0.03099	0.02010	0.00889
50,000	0.12243	0.10657	0.02833
100,000	0.24124	0.18758	0.04096
500,000	1.56116	1.06417	0.20375
1,000,000	3.66643	2.41001	0.42598
5,000,000	22.88610	13.38121	2.55142
10,000,000	47.63483	29.97095	5.19308
20,000,000	98.5066	55.0424	10.2748
30,000,000	136.2943	81.3211	15.7475
40,000,000	199.6229	110.9872	22.8183
50,000,000	254.68691	145.26292	31.7804

Table 6
The actual running times (in seconds) of the algorithms for circle data.

Input	$\mathcal{O}$-Graham	Montuno & Fournier's algorithm	$\mathcal{O}$-Quickhull
1000	0.0035	0.0023	0.0082
5000	0.0216	0.0170	0.0603
10,000	0.0376	0.0306	0.1279
50,000	0.1265	0.0976	0.4259
100,000	0.2199	0.1752	0.7540
500,000	1.4621	1.0193	3.7179
1,000,000	3.3078	2.2923	7.4452
5,000,000	20.8481	12.7159	35.8269
10,000,000	42.4375	27.3978	87.9717
20,000,000	97.0361	52.3828	156.8967
30,000,000	134.0237	79.6748	234.7828
40,000,000	187.0840	105.6379	346.9818
50,000,000	226.1866	133.2188	493.1219

Table 7
The average ratios between the actual running times in term of the geometric mean.

Data types	The ratio of $\mathcal{O}$-Graham to $\mathcal{O}$-QUICKHULL	The ratio of Montuno and Fournier's algorithm to $\mathcal{O}$-QUICKHULL
Square data	1.3364	1.2375
Hollow square data	1.2019	1.2326
Disc data	3.8311	2.3027
Hollow disc data	3.3321	2.1734
Hollow sun data	5.6741	3.6558
Circle data	0.4285	0.2864

Table 8
The ratios between regression parameters based on regression analysis.

Data types	$a_{\text{O-Graham}}/a_{\text{O-Quickhull}}$	$a_{\text{Montuno and Fournier}}/\ a_{\text{O-Quickhull}}$
Square data	1.1863	1.1197
Hollow square data	1.0853	1.2077
Disc data	5.0745	2.6391
Hollow disc data	4.1000	2.1696
Hollow sun data	8.3481	4.7438

of $O(n^2)$), and in this case, $\mathcal{O}$-QUICKHULL algorithm is slower than the other two ($\mathcal{O}$-Graham and Montuno and Fournier's algorithms have $O(n \log n)$ complexity.)

In Table 6, we list the average ratios between the actual running times of $\mathcal{O}$-Graham algorithm and Montuno and Fournier's algorithm versus $\mathcal{O}$-QUICKHULL for the test sets by geometric mean. Accordingly, the O-Quickhull algorithm is significantly faster than the other two algorithms in the first five data types.

Next, the regression method is also used to estimate the ratios between the actual running times of these algorithms. We first fit the data in Tables 1–5 via regression to curves of the form

$$runtime(input) \approx a_{\text{alg}}(input)\log(input),$$

Except for the data in the last column of Table 5, it is fit to a curve of the form

$$runtime(input) \approx a_{\text{alg}}(input)^2,$$

where *input* is the first column of each table, *runtime* is the algorithm depended running times provided in second column to fourth column of each table, and a_{alg} is the algorithm-dependent regression parameter. Then we calculate the ratio of the regression parameters (growth rates) for the best fit curves to estimate the ratio of the running times between the algorithms. In each of Tables 1–5, the fit curve for the data of each algorithm leads to the *convergence tolerance* less than 10^{-6}. And the fit curve for the data of the last column of Table 5 results in $r^2 > 0.99$. In Fig. 14, we add fit curves corresponding to the data in Tables 1–5. The ratios of the running times between the algorithms can be determined based on the growth rates of these fit curves. The results of the first five data types are presented in Table 7.

6. Concluding remarks

We have provided $\mathcal{O}$-QUICKHULL algorithm inspired by the idea of the Quickhull algorithm, for finding the c.o. convex hull and have compared it with the algorithm [7] and Montuno and Fournier's algorithm [24] with respect to the running time. A similar algortihm for finding $\mathcal{O}_\beta$-convex hulls (introduced in [2]) can be given. In addition, we can use the idea of space subdivision [32] and the idea of the method of orienting curvers [5,6,28,29], to give efficient algorithms for finding such hulls. They will be the subject of another paper.

Acknowledgment

Nguyen Kieu Linh was funded by Vingroup JSC and supported by the Postdoctoral Scholarship Programme of Vingroup Innovation Foundation (VINIF), Institute of Big Data, code VINIF.2021.STS.18. The authors thank the Ho Chi Minh City University of Technology for the support of time and facilities. This research is funded byVietnam National University Ho Chi Minh City (VNU-HCM) under grant number NCM2021-20-02.

The authors would like to thank the referees for their insightful comments on the paper.

Appendix A

The proof of Proposition 2

Proof. Let E be a c.o. convex hull of P and F be the minimum rectangle having edges parallel to coordinates axises, where F is formed by $a, b, c, d \in P$. (see Fig. 15 (i)).

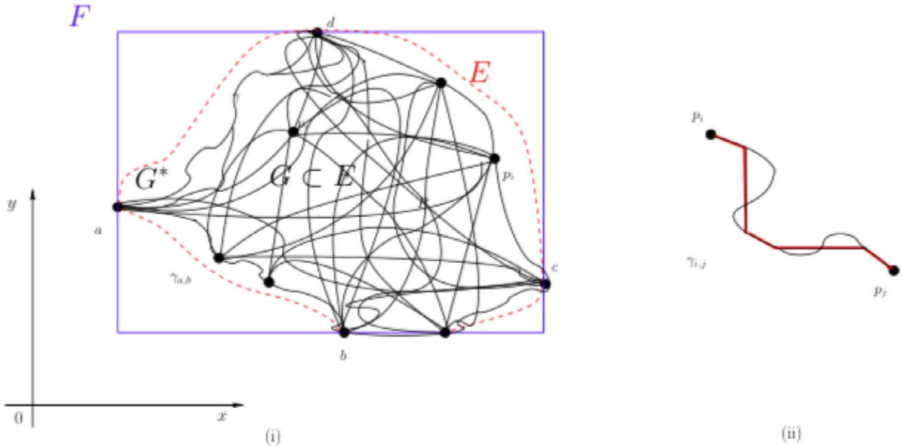


Fig. 15. (i) $G := \bigcup_{(i,j)\in[1,m]\times[1,m]} \gamma_{i,j}$ and the region G^* formed by the path γ. (ii) Adapting $\gamma_{i,j}$ to get an o. convex set.

We now claim that there is a compact o. convex subset of E containing P. By Proposition 1, for each pair $(i, j) \in [1, m] \times [1, m]$, exists a staircase path belonging to E, say $\gamma_{i,j}$, joining p_i and p_j(see Fig. 15 (ii)). Lemma 2 implies that $E \subset F$, then

$$G := \bigcup_{(i,j)\in[1,m]\times[1,m]} \gamma_{i,j} \subset F.$$

By the way, the closedness of $\gamma_{i,j}$ yields that G is closed. Thus G is compact. Let γ be the boundary of G. We prove that γ is a path. Let $\beta_1(x) := \min\{\gamma_{ij}(x) : (i, j) \in [1, m] \times [1, n]\}$, where $x \in [a_x, b_x]$. Since minimum of finite continuous functions is also continuous, $\beta_1(x)$ is continuous. Therefore, the part of γ from a to b is a path. By similar argument,

$$\beta_2(x) := \min\{\gamma_{ij}(x) : (i, j) \in [1, m] \times [1, n]\}, \text{where}\, x \in [b_x, c_x];$$
$$\beta_3(x) := \max\{\gamma_{ij}(x) : (i, j) \in [1, m] \times [1, n]\}, \text{where}\, x \in [b_x, c_x];$$
$$\beta_4(x) := \max\{\gamma_i(x) : (i, j) \in [1, m] \times [1, n]\}, \text{where}\, x \in [a_x, b_x]$$

are also continuous. Therefore, the parts of γ from b to c (β_2), from c to d (β_3) and from d to a (β_4) are paths. Then γ is a path. By the way chosen each part of γ, we get γ is not self-cross. Thus γ bounds a region G^*. We have $G^* \subset E$ and G^* is connected and contains P.

We are in position to prove that G^* is o. convex. Take a rectilinear line k intersecting G^*. Let $u, v \in k \cap G^*$ being two "farthest" points which still lie in G^*. Assume without loss of generality that $[u, v]$ is parallel to x-axis. We claim that $[u, v] \subset G^*$. Assume the contrary that $z \in [u, v] \setminus G^*$ (see Fig. 16 (i)). Consider the case u belongs to the part $\gamma_{a,b}$ of γ between a and b and $a_x < z_x < b_x$ (the other cases are similar). As $\gamma_{a,b}$ is formed by some monotone paths joining two points of P, there are three points $g, h, l \in P$ such that h is above $[u, v]$, g, l are under $[u, v]$, $\gamma_{h,g}$ and $\gamma_{h,l}$ are monotone (see Fig. 16 (ii)). Since g, l are under $[u, v]$, the monotone path $\gamma_{g,l}$ is under $[u, v]$. Therefore z belongs to the region formed by $\gamma_{g,l}$, $\gamma_{h,g}$ and $\gamma_{h,l}$. This implies that $z \in G^*$, a contradiction. Thus, G^* is o. convex.

Because E is the smallest c.o. convex set containing P, we conclude that $G^* = E$. Thus E is compact. □

The proof of Lemma 4

Proof. Taking $p \in \mathrm{COCH}(P)$, we consider three cases: $p \in \mathrm{ext}(\mathrm{COCH}(P))$, p is on the boundary but $p \notin \mathrm{ext}(\mathrm{COCH}(P))$ (see Fig. 17) and p is in the interior of $\mathrm{COCH}(P)$ (see Fig. 18).

Case 1: $p \in \mathrm{ext}(\mathrm{COCH}(P))$. Then clearly all four orthants of p contain the extreme point p of $\mathrm{COCH}(P)$.

Case 2: p is on the boundary but $p \notin \mathrm{ext}(\mathrm{COCH}(P))$. According to Lemma 3, p must belong to an o. support $\mathrm{l}(a, b)$ that goes through two extreme points a and b of $\mathrm{COCH}(P)$. If o. support $\mathrm{l}(a, b)$ is a straight line (i.e, $x_a = x_b$ or $y_a = y_b$), then clearly two orthants of p contain the extreme point a and the remaining two orthants contain the extreme point b (see Fig. 17 i)).

As two some orthants of p contain the extreme point a, without loss of generality assume that p belongs to the rectilinear half line containing b (see Fig. 17 ii)). Then two orthants of p contain the extreme point b. One of the remaining two orthants contains the extreme point a. We will show that the final orthants of p (e.g., $o_4(p)$ as shown in Fig. 17 ii)) contain at least one other extreme point of $\mathrm{COCH}(P)$. Indeed, assume that $o_4(p)$ does not contain any points of $\mathrm{ext}(\mathrm{COCH}(P))$. Then, by Lemma 1(ii), $o_4(p)$ does not contain any point of P. Let u be a smallest y-coordinate point among the points of P in $o_1(p)$, and v be a greatest x-coordinate point among the points of P in $o_3(p)$ (since the set P is finite, u and v exist). Therefore an o. line $\mathrm{l}(u, v)$ is an o. support and $\mathrm{l}(u, v)$ intersects $\mathrm{l}(a, b)$ at exactly two distinct points. Thus, the c.o. convex hulls of P have semi-isolated points. This is contrary to assumption (A).

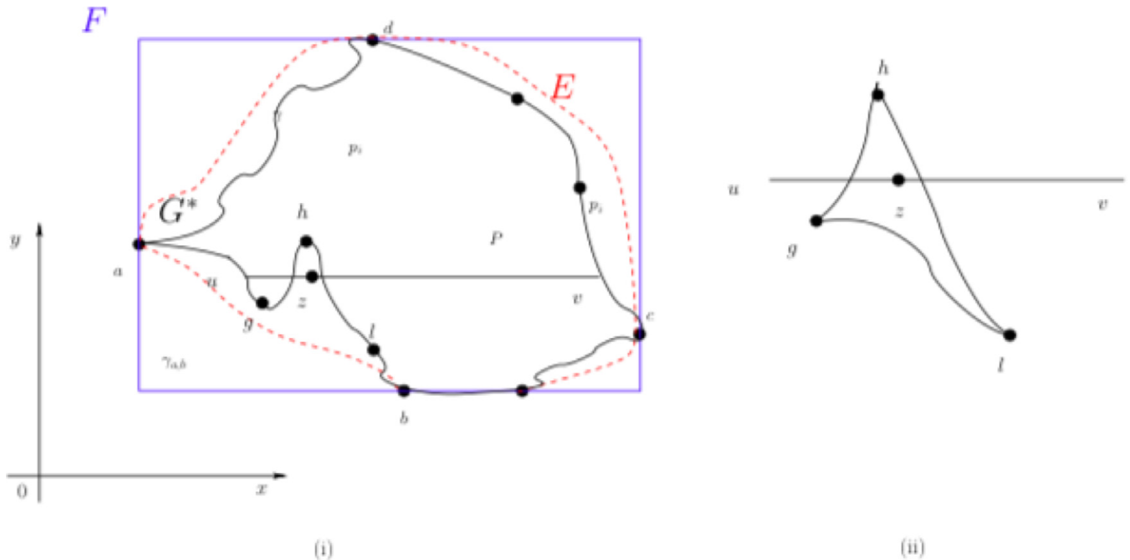


Fig. 16. $z \in [u, v]$ in the region G^* formed by the monotone paths $\gamma_{h,g}$, $\gamma_{h,l}$ and $\gamma_{g,l}$.

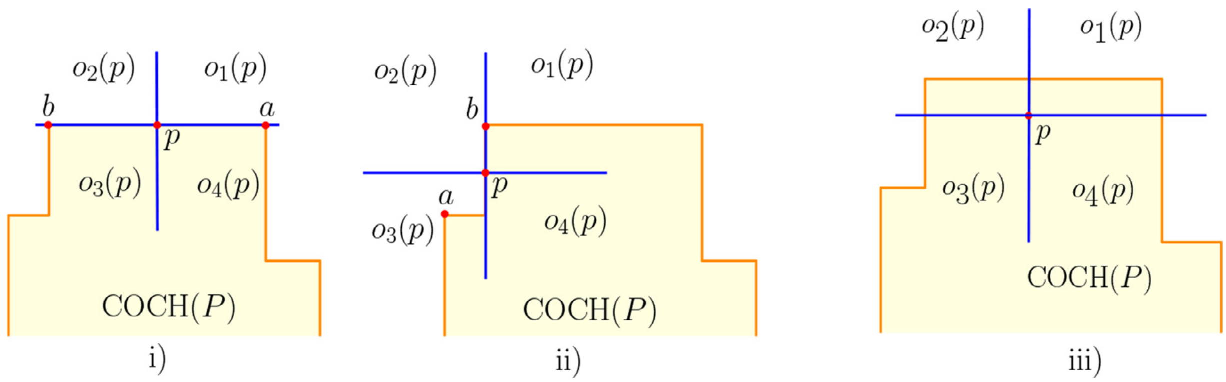


Fig. 17. p lies on the boundary of COCH(P).

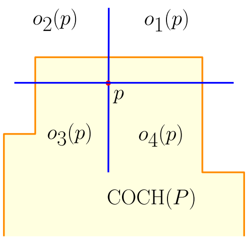


Fig. 18. p is in the interior of COCH(P).

Case 3: p is in the interior of COCH(P). Then the orthants $o_1(p)$, $o_2(p)$, $o_3(p)$, and $o_4(p)$ intersect COCH(P) $\setminus$ {p}. Applying the similar argument for $o_4(p)$ in Case 2 to these orthants, we conclude that each orthant contains at least one extreme point of COCH(P). □

The proof of Lemma 5

Proof. Let $p \in \mathcal{M}(P)$, there is an its orthant which contains no points of $P \setminus \{p\}$. Without loss of generality we assume that $o_2(p)$ does not contain any points of $P \setminus \{p\}$. We will prove that $o_2(p)$ does not contain any points of COCH$(P) \setminus \{p\}$. We will prove this by contradiction. Indeed, suppose that there exists $u \in o_2(p) \cap (\text{COCH}(P) \setminus \{p\})$. There are two following cases:

- $u \in \text{ext}(\text{COCH}(P))$. According to Lemma 1(i), $u \in P$, namely $o_2(p)$ contains a point$u \in P \setminus \{p\}$.

- $u \notin$ ext(COCH(P)). According to Lemma 4, each orthant of u contains at least one point of ext(COCH(P)). Therefore, $o_2(u)$ also contains at least one extreme point, say t, of COCH(P) and according to Lemma 1(i), $t \in P$. Furthermore, since $u \in o_2(p)$, we have $o_2(u) \subseteq o_2(p)$. It follows that $o_2(p)$ contains the point $t \in P \setminus \{p\}$.

Both cases above contradict the hypothesis that $o_2(p)$ does not contain any points of $P \setminus \{p\}$. Hence, $o_2(p)$ does not contain any points of COCH(P) $\setminus \{p\}$, i.e., $p \in$ ext(COCH(P)). Therefore

$$\mathcal{M}(P) \subseteq \text{ext}(\text{COCH(P)}). \tag{6}$$

Conversely, let $p \in$ ext(COCH(P)), now we will prove that $p \in \mathcal{M}(P)$. Indeed, Lemma 1(i) implies $p \in P$. According to Definition 5, there exists an orthant $o(p)$ of p that does not contain any points of COCH(P) $\setminus \{p\}$. We can conclude from Lemma 1(ii) that $o(p)$ also does not contain any points of $P \setminus \{p\}$, i.e., $p \in \mathcal{M}(P)$. Therefore

$$\text{o-ext}(\text{COCH}(P)) \subseteq \mathcal{M}(P). \tag{7}$$

It follows from (6) and (7) that $\mathcal{M}(P) = \text{ext}(\text{COCH}(P))$. □

References

[1] J. Abello, V. Estivill-Castro, T. Shermer, J. Urrutia, Illumination of orthogonal polygons with orthogonal floodlights, Int. J. Comput. Geom. Appl. 8 (1998) 25–38.
[2] C. Alegría, D. Orden, C. Seara, J. Urrutia, On the $\mathcal{O}$-hull of a planar point set, Comput. Geometry 68 (2018) 277–291.
[3] C. Alegría, D. Orden, C. Seara, J. Urrutia, Efficient computation of minimum area rectilinear convex hull under rotation and generalizations, J. Glob. Optim. 79 (2021) 687–714.
[4] M.A.A.A. Aman, A. Sarkar, M. Dutt, A. Biswas, A linear time combinatorial algorithm to compute the relative orthogonal convex hull of digital objects, Theor. Comput. Sci. 847 (2020) 103–1121.
[5] P.T. An, Method of orienting curves for determining the convex hull of a finite set of points in the plane, Optimization 59 (2) (2010) 175–179.
[6] P.T. An, Finding shortest paths in a sequence of triangles in 3d by method of orienting curves, Optimization 67 (2018) 159–177.
[7] P.T. An, P.T.T. Huyen, N.T. Le, A modified grahams scan algorithm for finding the smallest connected orthogonal convex hull of a finite planar point set, Appl. Math. Comput. 397 (2021) 125889.
[8] P. Balázs, On the number of hv-convex discrete sets, in: V.E. Brimkov, R.P. Barneva, H.A. Hauptman (Eds.), Combinatorial Image Analysi, IWCIA, Lecture Notes in Computer Science, 4958, Springer, 2008.
[9] T. Biedl, B. Genc, Reconstructing orthogonal polyhedra from putative vertex sets, Comput. Geom. 44 (2011) 409–417.
[10] F. Bookstein, Morphometric Tools for Landmark Data: Geometry and Biology, Cambridge University Press, 1991.
[11] L. Boxer, Computing deviations from convexity in polygons, Pattern Recognit. Lett. 14 (1993) 163–167.
[12] A. Bykat, Convex hull of a finite set of points in two dimensions, Inf. Process. Lett. 7 (1978) 296–298.
[13] L.D.F. Costa, J.R.M. Cesar, Shape Analysis and Classification, CRC Press, 2001.
[14] V. Damerow, C. Sohler, Extreme points under random noise, Eur. Symp. Algo. 3221 (2004) 264–274.
[15] W. Eddy, A new convex hull algorithm for planar sets, ACM Trans. Math. Softw. 4 (1977) 398–403.
[16] E. Fink, D. Wood, Fundamentals of restricted-orientation convexity, Inf. Sci. 92 (1996) 175–196.
[17] H. Gonzalez-Aguilar, D. Orden, P.L. Pablo, R. David, S. Carlos, T. Javier, U. Jorge, Maximum rectilinear convex subsets, Int. Symp. Fundam. Comput. Theory (2019) 274–291.
[18] N.D. Hoang, N.K. Linh, Quicker than quickhull, Vietnam J. Math. 43 (2015) 57–70.
[19] S.T. Hyde, S. Andersson, Z. Blum, S. Lidin, K. Larsson, T. Landh, B.W. Ninham, The Language of Shape, Elsevier, 1997.
[20] Y. Gerard, Regular switching components, Theor. Comput. Sci. 777 (2019) 338–355.
[21] R.G. Karlsson, M.H. Overmars, Scan line algorithms on a grid, BIT Numer. Math. 28 (1988) 227–241.
[22] N.K. Linh, C. Song, J. Ryu, P.T. An, N.D. Hoang, D.S. Kim, Quickhulldisk: a faster convex hull algorithm for disks, Appl. Math. Comput. 363 (2019) 124626.
[23] P. Mitura, I. Simecek, I. Kotenkov, Effective construction of convex hull algorithms, in: 19th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Timisoara, IEEE, 2017, pp. 105–112.
[24] D.Y. Montuno, A. Fournier, Finding the X-Y Convex Hull of a Set of X-Y Polygons, 148, Technique Report Uni, Toronto, 1982.
[25] T.M. Nicholl, D.T. Lee, Y.Z. Liao, C.K. Wong, On the x-y convex hull of a set of x-y polygons, BIT Numer. Math. 23 (1983) 456–471.
[26] J. O'Rourke, Computational Geometry in C, Cambridge University Press, 1998.
[27] T. Ottmann, E. Soisalon-Soininen, D. Wood, On the definition and computation of rectilinear convex hulls, Inf. Sci. 33 (1984) 157–171.
[28] H.X. Phu, Ein konstruktives lösungsverfahren für das problem des inpolygons kleinsten umfangs von J. Steiner, Optimization 18 (1987) 349–359.
[29] H.X. Phu, Zur lösung eines zermeloschen navigationsproblems, Optimization 18 (1987) 225–236.
[30] F. Preparata, M. Shamos, Computational Geometry, Springer, 1985.
[31] J. Seo, S. Chae, J. Shim, D. Kim, C. Cheong, T. Han, Fast contour-tracing algorithm based on a pixel-following method for image sensors, Sensors 16 (2016) 353.
[32] V. Skala, Diameter and Convex Hull of Points Using Space Subdivision in $\dot{e}^2$ and $\dot{e}^3$, gervasi o. et al, Computational Science and Its Applications, Lecture Notes in Computer Science, 12249, Springer, 2020.
[33] W. Son, S.W. Hwang, H.K. Ahn, MSSQ: manhattan spatial skyline queries, Inf. Syst. 40 (2014) 67–83.
[34] M. Sonka, V. Hlavac, R. Boyle, Image Processing Analysis and Machine Vision, Chapman and Hall, 1993.
[35] H.I. Stern, Polygonal entropy: a convexity measure, Pattern Recognit. Lett. 10 (1989) 229–235.
[36] W. Trutschnig, G. Gonzlez-Rodrguez, A. Colubi, M.A. Gil, A new family of metrics for compact convex (fuzzy) sets based on a generalized concept of mid and spread, Inf. Sci. 179 (2009) 3964–3972.
[37] E. Uchoa, M.P. de Aragao, C.C. Ribeiro, Preprocessing steiner problems from VLSI layout, Networks 40 (2002) 38–50.
[38] S.H. Unger, Pattern detection and recognition, Proc. Inst. Radio Eng. IRE, IEEE (1959) 1737–1752.
[39] J. Wu, Z. Jiang, On constructing the minimum orthogonal convex polygon for the fault-tolerant routing in 2-d faulty meshes, IEEE Trans. Reliab. 54 (2005) 449–458.
[40] J. Zunic, P.L. Rosin, A new convexity measure for polygons, IEEE Trans. PAMI 26 (2004) 923–934.