JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 40, 865-876 (2024)
DOI: 10.6688/JISE.202407_40(4).0011

Separating Bichromatic Point Sets by an Axis-Parallel *C*-Shaped Polygon

FATEMEH KESHAVARZ-KOHJERDI[1,+], ANITA SHEYBANI[2] AND ALIREZA BAGHERI[2]
[1]*Department of Computer Science*
Shahed University
Tehran, 3319118651 Iran
E-mail: f.keshavarz@shahed.ac.ir
[2]*Department of Computer Engineering*
Amirkabir University of Technology (Tehran Polytechnic)
Tehran, 1591634311 Iran
E-mail: {sheybani.anita; ar_bagheri}@aut.ac.ir

In the separability problem of bichromatic point sets, given two sets of points colored as blue and red, we want to put a special geometric shape in a manner that includes the blue points while avoiding the red ones. This problem has various applications in data mining and other fields. Separability by various shapes, including *L*-shaped polygons, has been studied in the literature. In this paper, the separability of bichromatic point sets by *C*-shaped polygons, which are more general than *L*-shaped polygons, is studied and an $O(n \log n)$-time algorithm is presented, where n is the total number of points.

Keywords: computational geometry, separability, bichromatic point sets, *C*-shaped polygon, covering

1. INTRODUCTION

The *covering problem* is one of the important and widely used problems in the field of computational geometry, which has a great variety. In this problem, n fixed points on the plane and a geometric object is given. The goal is to put the geometric object on the plane that covers all the given points. According to optimization factors, such as minimizing the area or the perimeter of the covering shapes, different variations of the problem such as complete covering and optimal covering are raised. Due to the variety of geometric shapes (covering objects), such as convex hulls [13], circles [10], strips [7], triangles [3,14], rectangles [19] and *L*-shaped polygons [2] various covering problems are proposed and solved. Another variation of the covering problem is covering bichromatic points, which is also known as *separability* or *separation*, in which the given points are divided into two categories of desirable and undesirable points. Usually the desirable points are shown in blue and the undesirable points in red. In this problem, the goal is to cover the desired (blue) points by the given geometric shape (separating object) such that none of the undesirable (red) points are covered. Optimization factors such as

Received January 15, 2023; revised June 27, 2023; accepted July 21, 2023.
Communicated by Shi-Chun Tsai.
+ Corresponding author.

minimizing the perimeter and the area may also be considered for the separating object. In the following, some related works are mentioned.

One of the first separating problems is the separation of points by a line, in which the goal is to find a line with blue points on one side and red points on the other side. To determine the separability of points by a line, an algorithm with $O(n)$ time presented [10]. Also finding all the separating lines in $O(n)$ time is possible [16]. Separation of points by separating strips is another important separation problem. In this problem, the goal is to find two parallel lines (a strip) such that all the blue points lie between the two lines and the red points lie outside them. An algorithm with time complexity $O(n\log n)$ was proposed to solve this problem [8] and then it was proved that this time is optimal [1]. If the set of points are separable by a strip, reporting the angles of inclination of the separating strips in $O(n\log n)$ time is possible. If all the possible angles are computed before, then it is possible to find the strip of the minimum width and the strip of the maximum width, in times $O(n)$ and $O(n\log n)$, respectively [8, 16]. As one of the other variations we can mention the separating by wedges [8, 16]. Separability by a circle can be checked in $O(n)$ time [15]. The separation of points by convex polygons of the minimum number of edges is another problem raised in this field. Reporting a convex polygon of the minimum number of edges or reporting its absence is possible at the optimal time $O(n\log n)$ [5, 8]. In 2009, Kreveld *et al*. proposed the separation of points by a rectangle, and proved that reporting angles for which there is a separating rectangle is possible in $O(n\log n)$ time [9]. They also asked to investigate the problem when the separating object is a non-convex geometric shape. The separability of points by L-shaped polygons studied in [17]. Then, it was followed by some variations of the problem [17, 18]. The separability by two disjoint parallel rectangles is investigated in two cases, when they are parallel to the coordinate axes and when they are not, and two algorithms of time complexity $O(n\log n)$ and $O(n^2\log n)$ are presented, respectively [11]. Some algorithms with $O(n\log n)$ time are proposed to separate red and blue points by triangles with a fixed angle [12].

The separability problem has many applications in image processing, pattern recognition, statistical calculations and data mining [6]. For example, in the classification problem, which is a well-known problem in data mining field, the points with different properties must be separated and categorized into different classes. Separability by different shapes including triangles, circles, rectangles, and L-shaped polygons has studied in the literature. In this paper, the problem of separating bichromatic point sets by axis-parallel C-shaped polygons, which are more general than L-shaped polygons, is studied and an algorithm with $O(n\log n)$ time is presented, where n is the total number of points.

The structure of the paper is as follows. Some definitions are given in Section 2. In Section 3, the necessary and sufficient conditions for separating of the points by a C-shaped polygon are given. The conclusion is given in Section 4.

2. PRELIMINARIES

In this section, we provide some definitions that will be used throughout this paper. As mentioned in the introduction, in the separability problem there are two sets of desirable and undesirable points, which are colored by blue and red, respectively. Let the set of blue points be denoted by Q and the set of red points be denoted by P. Throughout

the paper, we assume that no two points (blue and red) have equal x or y coordinates. A *90-degree wedge* (or *quarter*) is defined by the intersection of two half-planes such that its supporting lines are axis-aligned and form a 90-degree angle. If there is no point of set Q inside this quarter, then it is called an empty quarter of Q. The *orthogonal convex hull* is defined as $RCH(Q) = \mathbb{R}^2 - \bigcup q$, where q is an empty quarter of Q [2, 4]. Let $R(Q)$ be the minimum area axis-parallel rectangle containing the blue points Q. Clearly, there is a blue point on each edge of $R(Q)$, note that a blue point may be shared by two edges of $R(Q)$. Consider the vertical line passing through the highest point of $R(Q)$, u_1 in Fig. 1 (a), and the horizontal line passing through the rightmost point of $R(Q)$, u_2 in Fig. 1 (a). In the figures, the red points are shown by black color, and the blue points are shown by gray color. The intersection area of the top half plane of this horizontal line and the right half plane of this vertical line is called *the first quarter* of $R(Q)$. The second, third, and fourth quarters of $R(Q)$ are defined similarly.

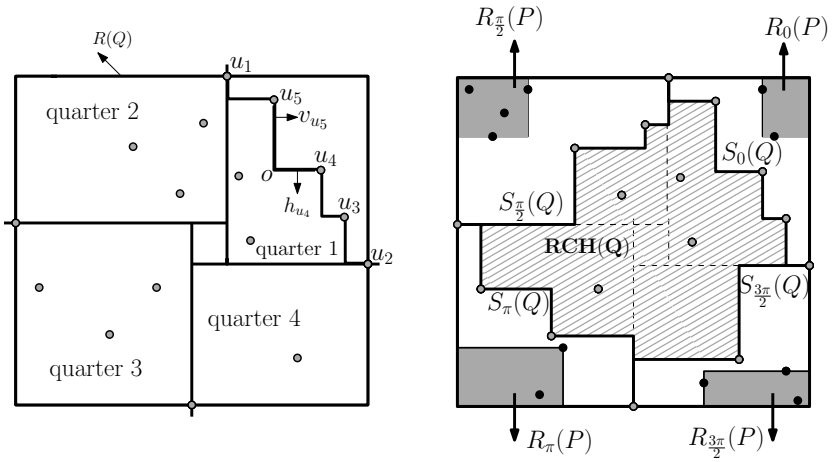


Fig. 1. (a) The four quarters of $R(Q)$ and the staircase of the first quarter and (b) $RCH(Q)$, $S_0(Q)$, $S_{\frac{\pi}{2}}(Q)$, $S_\pi(Q)$, and $S_{\frac{3\pi}{2}}(Q)$, and the red rectangles of the corners of $R(Q)$.

Let $p \in Q$ be in the first quarter of $R(Q)$ and x_p and y_p be the coordinates x and y of the point p, respectively. The point p is a *dominating point* if there is no point $q \in Q$ in the first quarter of $R(Q)$ such that $x_q > x_p$ and $y_q > y_p$. In Fig. 1 (a), the points u_1, u_2, u_3, u_4, and u_5 are the dominating points. Clearly based on the x or y coordinates of the points, we can define a total order δ on the dominating points. This definition is defined similarly for the second, third, and fourth quarters of $R(Q)$. In the following, the definitions are presented for the first quarter and can be extended to the other quarters.

Two dominating points a and b are *adjacent* if they are consecutive and there is no point between them in the total order δ. In Fig. 1 (a), u_1 is adjacent to u_5, u_5 is adjacent to u_4, u_4 is adjacent to u_3, and u_3 is adjacent to u_2. Suppose u_4 and u_5 are two adjacent dominating points, where $x_{u_5} < x_{u_4}$. Consider the vertical line passing through u_5 and the horizontal line passing through u_4. As shown in Fig. 1 (a), these two lines intersect at a point o. The vertical line segment (u_5, o) and the horizontal line segment (u_4, o) form a *step*. If we draw the vertical and the horizontal line segments of each pair of adjacent

dominating points, then a chain is formed which we call a *staircase* (Fig. 1 (a)). The chain $(u_1, u_5, u_4, u_3, u_2)$ is the staircase of the first quarter. Similarly, the staircases of the other quarters are specified.

Assume that $S_0(Q)$, $S_{\frac{\pi}{2}}(Q)$, $S_\pi(Q)$, and $S_{\frac{3\pi}{2}}(Q)$ are the staircases corresponding to the dominating points of the first, second, third, and fourth quarters of $R(Q)$, respectively. In addition to the previous definition of the orthogonal convex hull $RCH(Q)$, it is also defined by the four staircases $S_0(Q)$, $S_{\frac{\pi}{2}}(Q)$, $S_\pi(Q)$, and $S_{\frac{3\pi}{2}}(Q)$. See Fig. 1 (b). Consider the red points within the first quarter of $R(Q)$. From these points, imagine the ones that are outside of $RCH(Q)$. Consider the minimum area axis-parallel rectangle MAR_0 covering these points. Let $R_0(P)$ be an axis-parallel rectangle whose top-right corner is the same as the top-right corner of $R(Q)$ and whose bottom-left corner is the same as the bottom-left corner of MAR_0 (the red rectangle on the top-right corner of $R(Q)$, see Fig. 1 (b)). Similarly, the rectangles $R_{\frac{\pi}{2}}(P)$, $R_\pi(P)$, and $R_{\frac{3\pi}{2}}(P)$ are defined in the second, third and fourth quarters of $R(Q)$, respectively. These rectangles, *i.e.* $R_0(P)$, $R_{\frac{\pi}{2}}(P)$, $R_\pi(P)$, and $R_{\frac{3\pi}{2}}(P)$, are called the red rectangles of the corners of $R(Q)$.

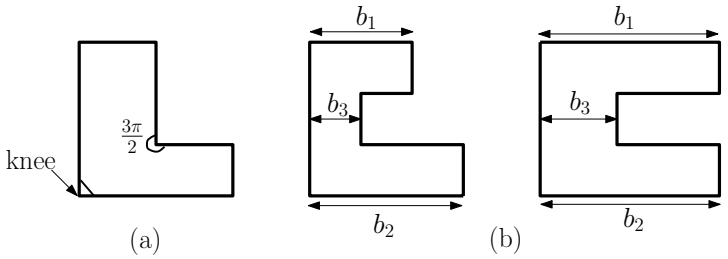


Fig. 2. *L*-shaped and *C*-shaped polygons.

An *L-shaped polygon* is a polygon that resembles the letter *L*. Fig. 2 (a) shows an *L*-shaped polygon. In an *L*-shaped polygon, the angle opposite to the interior angle of $\frac{3\pi}{2}$ is called the *knee*, see Fig. 2 (a). A *C-shaped polygon* is a polygon that resembles the letter *C*. Fig. 2 (b) shows a *C*-shaped polygon. Note that we may have $b_1 = b_2$ or $b_1 \neq b_2$ ($b_1 > b_3$ and $b_2 > b_3$). In the following, let $C(Q)$ denote the separating *C*-shaped polygon, which contains all the blue points of Q and excludes all the red points of P. The edges of a *C*-shaped polygon, in general, may not be axis-parallel, but we only consider axis-parallel *C*-shaped polygons.

3. COMPUTING THE SEPARATING *C*-SHAPED POLYGON

In this section, we give the necessary and sufficient conditions for separability of the given bichromatic points by an axis-parallel *C*-shaped polygon. The general idea of the proposed method is to compute the bounding rectangle $R(Q)$ of the blue points and cover the red points inside it by a rectangle R or an *L*-shaped polygon L. Then $R(Q) \setminus R$ or $R(Q) \setminus L$ is the separating *C*-shaped polygon. Suppose $RCH(Q)$, $R(Q)$, $R_0(P)$, $R_{\frac{\pi}{2}}(P)$, $R_\pi(P)$, and $R_{\frac{3\pi}{2}}(P)$ have been already computed. Even if $RCH(Q)$ is bichromatic, the points may be separable by a *C*-shaped polygon. But it is clear that for separability by

a C-shaped polygon, red points cannot be anywhere freely inside $RCH(Q)$ and some conditions must hold on their positions.

Lemma 3.1 *A necessary condition for separability by a C-shaped polygon $C(Q)$ is that at least three of the four red rectangles of the corners of $R(Q)$ be empty.*

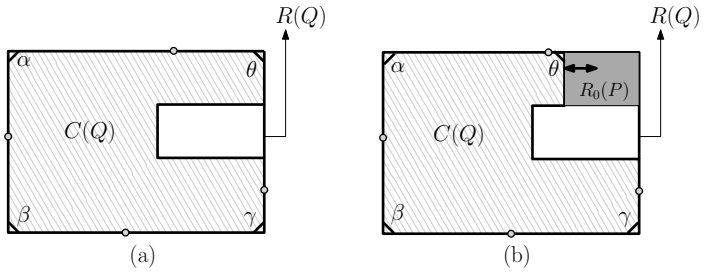


Fig. 3. Separability by a C-shaped polygon.

***Proof*:** As mentioned before, there is at least one blue point on each edge of $R(Q)$. Note that a blue point may be shared between two incident edges of $R(Q)$. The blue points on the edge of $R(Q)$ must be covered by $C(Q)$, so $C(Q)$ should be in one of the two forms as shown in Fig. 3. As can be seen in Fig. 3, this C-shaped polygon has at most four corners in common with $R(Q)$, the top-right corner θ, the bottom-right corner γ, the top-left corner α, and the bottom-left corner β. From these four corners of $C(Q)$, the three corners α, β, and γ have blue points on their both edges and there is exactly one corner θ that may have no blue point on one of its edges. Therefore, this edge has the freedom to move. This freedom means that because there is no blue point on this edge and it is not fixed by any blue point, it can move in parallel to itself (see Fig. 3 (b)). If some red points are in a corner of $R(Q)$, then that corner must be the θ corner of $C(Q)$. Hence, at least three of the four corners of $R(Q)$ must be empty of red points. ■

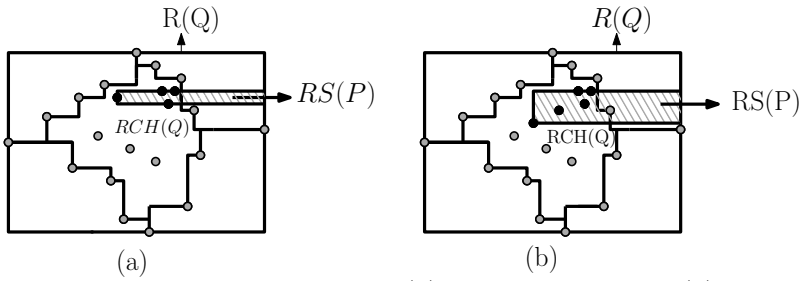


Fig. 4. (a) A monochromatic $RS(P)$ and (b) a bichromatic $RS(P)$.

Let $RS(P)$ denote a rectangle containing all the red points inside $RCH(Q)$ that is tangent to exactly one edge of the rectangle $R(Q)$ (see Fig. 4). By the length of an axis-parallel rectangle we mean the length of its horizontal edges.

Lemma 3.2 *If there is a separating C-shaped polygon $C(Q)$, then a monochromatic rectangle $RS(P)$ (containing only red points) there exists. Also when exactly one of the red rectangles of the corners of $R(Q)$, say $R_0(P)$, is non-empty then a monochromatic L-*

shaped polygon $L(P)$ that contains $R_0(P)$ and the rectangle $RS(P)$ there exists. In this case, $R_0(P)$ and $RS(P)$ are both tangent to the same edge of $R(Q)$, say the right edge of $R(Q)$. Also the length of $RS(P)$ is greater than the length of $R_0(P)$.

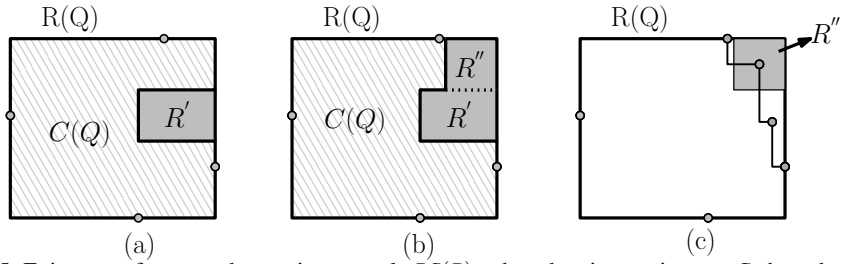


Fig. 5. Existence of a monochromatic rectangle $RS(P)$, when the given points are C-shaped separable.

***Proof*:** We know that there is a blue point on every edge of the rectangle $R(Q)$. Also, according to Lemma 3.1, $R(Q)$ can have at most one non-empty corner red rectangle. Therefore, if there is a separating C-shaped polygon $C(Q)$, it may have been constructed in one of the two ways as shown in Figs. 5 (a) and (b). In way (i), see Fig. 5 (a), there is no non-empty red rectangle at the corners of $R(Q)$. Let $R^{'} = R(Q) \setminus C(Q)$, which is a rectangle that has no blue points. In way (ii), see Fig. 5 (b), there is exactly one non-empty red rectangle at a corner of $R(Q)$, say $R_0(P)$. Let $L(P) = R(Q) \setminus C(Q)$, which is an L-shaped that has no blue points. By a cut, we partition this L-shaped into two rectangles $R^{'}$ and $R^{''}$. If the knee of the L-shaped is on the horizontal (resp. vertical) edge of $R(Q)$, then we make a vertical (resp. horizontal) cut. The rectangle that intersects the corner of $R(Q)$ is called $R^{''}$ and the other one is called $R^{'}$ (see Fig. 5 (b)). By the definition of $RCH(Q)$, $R^{''}$ can have nothing in common with $RCH(Q)$, otherwise a blue point lies inside $R^{''}$ (5 (c)). While we know that $R^{''}$, which is a part of the L-shaped, is empty of blue points. Therefore, all the red points inside $RCH(Q)$ that are separated by the C-shaped polygon from the blue points, are all inside $R^{'}$. By the definition of $R^{'}$, there is no blue point inside it, and has exactly on one edge tangent to $R(Q)$. So we can set $RS(P) = R^{'}$. Hence a monochromatic $RS(P)$ there exists. In way (ii), as can be seen in Fig. 5 (b), the L-shaped polygon $L(P)$ contains $R_0(P)$ and $RS(P)$. Also, $R_0(P)$ and $RS(P)$ are both tangent to the right edge of $R(Q)$. Further, in polygon $L(P)$, the rectangle $R_0(P)$ is included in $R^{''}$. So, the length of rectangle $RS(P) = R^{'}$ is greater than the length of rectangle $R_0(P)$. ■

Theorem 3.3 *If the necessary conditions for separability by a C-shaped polygon $C(Q)$ mentioned in Lemmas 3.1–3.2 hold, then there is a separating C-shaped polygon $C(Q)$, which means that these conditions are sufficient.*

***Proof*:** If the four red rectangles at the corners of $R(Q)$ are empty, according to Lemma 3.2 there is a monochromatic rectangle $RS(P)$, then it is clear that $C(Q) = R(Q) \setminus RS(P)$ is a separating C-shaped polygon. If exactly one of the four red rectangles at the corners of $R(Q)$ is not empty, according to Lemma 3.2 there is a monochromatic L-shaped polygon $L(P)$, then it is clear that $C(Q) = R(Q) \setminus L(P)$ is a separating C-shaped polygon. ■

For an axis-parallel rectangle R, then by $y_{min}(R)$ and $y_{max}(R)$ we mean y coordinates

Algorithm 3.1: The algorithm for computing a separating C-shaped polygon

Input: bichromatic point sets P and Q
Output: an axis-parallel separating C-shaped polygon $C(Q)$

1: Compute the min and the max of x and y coordinates of the points of Q, then compute $R(Q)$
2: Use the min and the max x and y of Q to compute the four quarters of blue points of Q
3: For each quarter of Q compute the related staircase $S_\theta(Q)$, $\theta = 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$ as mentioned in Section 2, then compute the $RCH(Q)$. Staircases are stored in sorted arrays.
4: Remove the red points outside $R(Q)$
5: Compute $R_\theta(P)$, $\theta = 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$
6: Compute all candidates for $RS(P)$
7: **if** all $R_\theta(P)$ are empty **then**
8: **if** there is a monochromatic $RS(P)$ **then**
9: **return** $C(Q) = R(Q) \setminus RS(P)$
10: **end if**
11: **end if**
12: **if** only one of $R_\theta(P)$, say $R_0(P)$, is not empty **then**
13: Compute all candidates for $L(P)$
14: **if** there is a monochromatic $L(P)$ **then**
15: **return** $C(Q) = R(Q) \setminus L(P)$
16: **end if**
17: **end if**
18: **return** NULL

of the bottom-edge and the top-edge of R, respectively. Algorithm 3.1 shows the pseudo code for computing an axis-parallel separating C-shaped polygon. In the following, the steps of the algorithm and the time-complexity of each step are mentioned. First, in line 1 of Algorithm 3.1, in $O(n)$ time, the blue points of the minimum and the maximum x and y coordinates are found. Because we have computed the top, bottom, left, and right points of $R(Q)$ at the beginning, then $R(Q)$ can be computed in a constant time. In line 2 of Algorithm 3.1, the blue points inside each quarter of $R(Q)$ in $O(n)$ time are specified by comparing their coordinates to x and y coordinates of the four points top, bottom, left and right of $R(Q)$. In line 3 of Algorithm 3.1, we sort the blue points of each quarter according to y coordinates in $O(n \log n)$ time. Then by traversing this sorted list and comparing x coordinates of each point and its previous point, in $O(1)$ time for each point will determine whether this point is dominating. Therefore, in total in $O(n \log n)$ time the dominating points of each quarter are specified. We store the dominating points of each quarter in a sorted array. We also save the beginning and end points of each staircase separately in memory. Clearly, by specifying the above four staircases, $RCH(Q)$ is obtained.

In line 4 of Algorithm 3.1, the red points outside $R(Q)$ are removed in $O(n)$ time. To

do this, all points are compared to the four edges of $R(Q)$ and it is determined whether they are inside $R(Q)$ or not, which takes $O(1)$ time for each point. So, in total, for at most n red points, it is done in $O(n)$ time. In line 5 of Algorithm 3.1, we find all the red points outside $RCH(Q)$ and inside the first quarter in $O(n\log n)$ time. To determine if the points in the quarter are outside or inside $RCH(Q)$, they must be compared to the staircases of the quarter. The best time is obtained by a binary search on the sorted arrays of the staircase, which is $O(\log n)$ time for each point and $O(n\log n)$ time for the points of the quarter. Among these red points, the lowest point and the leftmost point indicate the bottom and left edges of the corner red rectangle $R_0(P)$, respectively. These two points are obtained in $O(n)$ time. Similarly, the other three corner red rectangles of $R(Q)$ are computed.

In line 6 of Algorithm 3.1 to compute $RS(P)$, first the red points inside $RCH(Q)$ are computed in $O(n\log n)$ time using arrays containing the first to fourth quarter staircase points. Then from the red points inside $RCH(Q)$, we compute the points with the minimum x and y coordinates in $O(n)$ time. Then the minimum rectangle containing the red points inside $RCH(Q)$, *i.e.* $MR(P)$, is computed in $O(1)$ time. By this rectangle, four candidate rectangles for $RS(P)$ are computed, which rises along the edges of the $MR(P)$ rectangle in four different directions top, bottom, left, and right. These four candidates are computed in $O(1)$ time.

In line 7 of Algorithm 3.1, checking whether these rectangles are empty (there exist or not) is done in $O(1)$ time. If the condition of line 7 of Algorithm holds, then in line 8 the checking monochromaticity of each of the four candidates for $RS(P)$ is computed in $O(n)$ time. If in line 8 of Algorithm 3.1 $RS(P)$ is monochromatic, then $C(Q) = R(Q) \setminus RS(P)$, otherwise returns NULL. Clearly $C(Q)$ is computed in $O(1)$ time. If the condition of line 7 of Algorithm 3.1 is not true, then in line 15 of the algorithm, if exactly one of the corner red rectangles of $R(Q)$ is not empty (say $R_0(P)$), then in line 16 of the algorithm we have to compute $L(P)$. From the four candidates for $RS(P)$, that candidate is selected to make $L(P)$ that is monochromatic and tangent to an edge of $R(Q)$ to which $R_0(P)$ is also tangent. It is possible to check these conditions for four candidates of $RS(P)$ in $O(1)$ time. For each candidates of $RS(P)$, there are three possible cases for constructing $L(P)$.

Case 1: If $y_{min}(RS(P)) \leqslant y_{min}(R_0(P)) \leqslant y_{max}(RS(P))$, then $RS(P)$ and $R_0(P)$ do not change.

Case 2: If $y_{max}(RS(P)) < y_{min}(R_0(P))$ (see Fig. 6 (a)), then the bottom edge of $R_0(P)$ extends (see Fig. 6 (b)).

Case 3: If $y_{min}(RS(P)) > y_{min}(R_0(P))$ (see Fig. 7 (a)), then bottom edge $RS(P)$ extends (see Fig. 7 (b)).

It is easy to see, if the length of $RS(P)$ is less than the length of $R_0(P)$ then the monochoromatic L-shaped $L(P)$ does not exist, and extension of the length of $RS(P)$ does not help. In all the three cases, $L(P)$ is formed by the union of the final $RS(P)$ and $R_0(P)$ (see Figs. 6 and 7). In this way, $L(P)$ is constructed in $O(1)$ time. Then in line 17 of Algorithm 3.1, the monochromaticity of $L(P)$ is checked in $O(n)$ time. If the condition of line 17 of the algorithm is true ($L(P)$ is monochromatic), then $C(Q) = R(Q) \setminus L(P)$ and it is returned, otherwise NULL is returned. If the condition of line 15 of the algorithm is not true, *i.e.* there exist more than one non-empty corner red rectangles, then NULL is

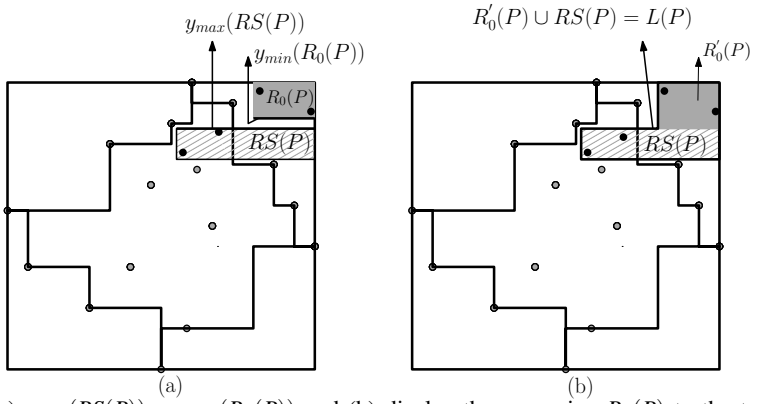


Fig. 6. (a) $y_{max}(RS(P)) < y_{min}(R_0(P))$ and (b) display the expansion $R_0(P)$ to the top edge of $RS(P)$.

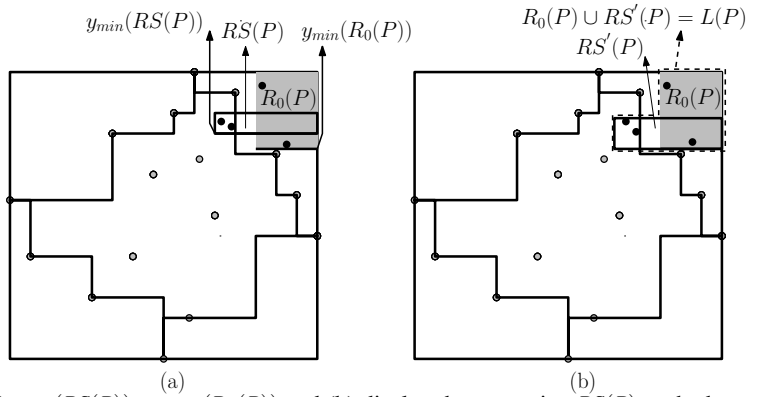


Fig. 7. (a) $y_{min}(RS(P)) > y_{min}(R_0(P))$ and (b) display the expansion $RS(P)$ to the bottom edge of $R_0(P)$.

returned. Clearly, $C(Q)$ is computed in $O(1)$ time-complexity. Therefore, the total time of the algorithm is $O(n \log n)$. Therefore, the following theorem is concluded.

Theorem 3.4 *Deciding whether two sets of points P and Q are C-separable can be decided in $O(n \log n)$ time. Moreover a C-separating polygon of P and Q can be computed in $O(n \log n)$ time and $O(n)$ space*

The given algorithm computes separating C-shaped polygon, which does not have necessarily the minimum area. To compute the minimum-area separating C-shaped polygon we should modify the algorithm. This can be done simply by extending the monochoromatic rectangle $RS(P)$ and L-shaped polygon $L(P)$ until their boundary touch the blue points. The number of separating C-shaped polygons may be unlimited in general. But the number of minimum-area separating C-shaped polygons is limited and we can enumerate all of them. This can be done by considering all the possible candidate monochoromatic rectangles $RS(P)$ and L-shaped polygons $L(P)$.

4. CONCLUSION AND FUTURE WORK

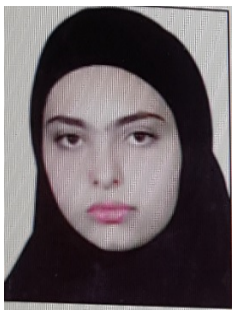
The set of blue points on the plane may not be separated from the red points by shapes that have been studied so far, such as a rectangle, two rectangles, or an L-shaped polygon, so it is necessary to use other shapes for separating. In this paper, the separability problem of bichromatic point sets by axis-parallel C-shaped polygons, which is more general than L-shaped polygons, is studied and an algorithm with a time complexity of $O(n \log n)$ is presented. As future work, we can consider more colors for the points.

REFERENCES

1. E. M. Arkin, F. Hurtado, J. S. B. Mitchell, C. Seara, and S. Skiena, "Some lower bounds on geometric separability problems," *International Journal of Computational Geometry & Applications*, Vol. 16, 2006, pp. 1-26.
2. S. W. Bae, C. Lee, H. K. Ahn, S. Choi, and K. Y. Chwa, "Computing minimum-area rectilinear convex hull and L-shape," *Computational Geometry: Theory and Application*, Vol. 42, 2009, pp. 903-912.
3. B. K. Bhathacharya and A. Mukhopadhyay, "On the minimum perimeter triangle enclosing a convex polygon," in *Proceedings of Japanese Conference on Discrete and Computational Geometry*, 2002, pp. 84-96.
4. J. Díaz-Báñez, C. Seara, J. Sellarès, J. Urrutia, and I. Ventura, "Covering point sets with two convex objects," in *Proceedings of European Workshop on Computational Geometry*, 2005, pp. 179-182.
5. H. Edelsbrunner and F. Preparata, "Minimum polygonal separation," *Information and Computation*, Vol. 77, 1988, pp. 218-232.
6. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning,* Springer-Verlag, Berlin, 2001.
7. M. E. Houle and G. Toussaint, "Computing the width of a set," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, 1988, pp. 761-765.
8. F. Hurtado, M. Noy, P. A. Ramos, and C. Seara, "Separating objects in the plane by wedges and strips," *Discrete Applied Mathematics*, Vol. 109, 2000, pp. 109-138.
9. M. V. Kreveld, T. V. Lankveld, and R. Veltkamp, "Identifying well-covered minimal bounding rectangles in 2D point data," in *Proceedings of the 25th European Workshop on Computational Geometry*, 2009, pp. 277-280.
10. N. Megiddo, "Linear time algorithm for linear programming in R^3 and related problems," *SIAM Journal of Computing*, Vol. 12, 1983, pp. 759-776.
11. Z. Moslehi and A. Bagheri, "Separating bichromatic point sets by two disjoint isothetic rectangles," *Scientia Iranica*, Vol. 23, 2016, pp. 1228-1238.
12. Z. Moslehi and A. Bagheri, "Separating bichromatic point sets by minimal triangles with a fixed angle," *International Journal of Foundations of Computer Science*, Vol. 28, 2017, pp. 309-320.
13. J. O'Rourke, *Computational Geometry in C*, 2nd ed., Cambridge University Press, UK, 1988.

14. J. O'Rourke, A. Aggarwal, S. Maddila, and M. Baldwin, "An optimal algorithm for finding minimal enclosing triangle," *Journal of Algorithms*, Vol. 7, 1986, pp. 258-269.
15. J. O'Rourke, S.R. Kosaraju, and N. Megiddo, "Computing circular separability," *Discrete Computational Geometry*, Vol. 1, 1986, pp. 105-113.
16. C. Seara, "Geometric on separability," Ph.D. Thesis, Department of Mathematics, University of Politecnica De Catalunya, 2002.
17. F. Sheikhi, A. Mohades, M. de Berg, and M. Davoodi, "Separating bichromatic point sets by *L*-shapes," *Computational Geometry: Theory and Applications*, Vol. 48, 2015, pp. 673-687.
18. F. Sheikhi, M. de Berg, A. Mohades, M. Davoodi, "Finding monochromatic *L*-shapes in bichromatic point sets," in *Proceedings of the 22nd Canadian Conference on Computational Geometry*, 2010, pp. 269-272.
19. G. Toussaint, "Solving geometric problems with the rotating calipers," in *Proceeding IEEE Mediterranean Electrotechnical Conference*, 1983, pp. 1-8.

Fatemeh Keshavarz-Kohjerdi received her Ph.D. degree in Computer Engineering from Amirkabir University of Technology, Tehran, Iran in 2016. She is presently an Assistant Professor of Computer Science at Shahed University, Tehran, Iran. Her current research interests include graph algorithms, robotics, computational geometry, and data mining.



Anita Sheybani received her BS degree in Computer Science from Alzahra University at Tehran. She received her MS degree in Computer Science from Amirkabir University of Technology at Tehran. Her research interest is computational geometry.

Alireza Bagheri received his BS and MS degrees in Computer Engineering from Sharif University of Technology at Tehran. He received his Ph.D. degree in Computer Science from Amirkabir University of Technology at Tehran. Currently he is an Associate Professor in Computer Engineering at Amirkabir University of Technology at Tehran. His research interests include computational geometry, graph algorithms, and social network analysis.