Minimizing the number of separating circles for two sets of points in the plane

Jiaye Wang
Department of Computer Science
Shandong Econimical University
Jinan, China
jywangz@sdu.edu.cn

Feng Sun
Department of Computer Science
The University of Hong Kong
Hong Kong, China
fsun@cs.hku.hk

Wenping Wang
Department of Computer Science
The University of Hong Kong
Hong Kong, China
wenping@cs.hku.hk

Chunyan Miao
School of Computer Engineering
Nanyang Technological University
Singapore
ascymiao@ntu.edu.sg

Caiming Zhang
Department of Computer Science
Shandong University
Jinan, China
czhang@sdu.edu.cn

***Abstract*—Given two sets of points $\mathbb{R}$ and $\mathbb{B}$ in the plane, we address the problem of finding a set of circles $\mathbb{C} = \{c_i, i = 1, 2, \ldots, k\}$, satisfying the condition that every point in $\mathbb{R}$ is covered by at least one circle in $\mathbb{C}$ and each point in $\mathbb{B}$ is not covered by any circle in $\mathbb{C}$. We conjecture that to find such a set with the smallest k is NP-hard. In this paper, we present an approximation algorithm for computing the set with minimal number of such circles. The algorithm finds also a lower bound of the smallest k.**

***Keywords*-separating circle, bichromatic points, Delaunay triangulation, separation, approximation algorithm**

I. Introduction

Separating two sets of primitives, such as points, line segments, by one or more separators, including hyperspheres and hyperplanes, has been studied in computational geometry for a long time. Boissonnat *et al.* [2] propose an algorithm for finding the largest circle separating two sets of line segments. Edelsbrunner and Preparata [4] bring up a method to find a convex polygon with minimum number of edges separating two given sets of points, if such polygons exist. Megiddo [8], [9], [10] proves that a hyperplane separating two sets of points can be found in linear time if it exists and proves that it is NP-hard to answer whether two sets of points in general space can be separated by two hyperplanes. In this paper, we will concentrate on the problem of separating two sets of points by circles.

The relationship between circles and points finds applications in wireless communication and motivates wide research in computational geometry. For a given set $\mathbb{S}$ of n points in the plane, Toussaint [12] presents an $O(n \log n)$ algorithm for computing a largest circle satisfying that its center locates in the convex hull and covers no points in $\mathbb{S}$; Megiddo [8] proposes an approach for finding the smallest enclosing circle of $\mathbb{S}$ in $O(n)$ time; de Berg *et al.* [3] present a randomized algorithm for finding the smallest enclosing circle of $\mathbb{S}$; this algorithm is easy to implement and has the expected time complexity $O(n)$. Another interesting problem is to find a smallest circle enclosing at least k points of a given set $\mathbb{S}$. Matoušek [7] presents a randomized algorithm with $O(n \log n + nk)$ expected time complexity. Har-Peled and Mazumdar [6] present an improved randomized algorithm with $O(nk)$ expected time complexity.

Given two sets of points $\mathbb{R}$ and $\mathbb{B}$ in the plane, one problem is to find a circle which contains all points in $\mathbb{R}$ and none of $\mathbb{B}$. Such a circle is called a *separating circle* for bichromatic points in the plane. Fisk [5] propose an $O(|\mathbb{R}||\mathbb{B}|)$ algorithm for computing centers and radii of all separating circles. O'Rourke *et al.* [11] finds the smallest separating circle in $O(n)$ time and the largest separating circle in $O(n \log n)$ time. When $\mathbb{R}$ is not completely separable from $\mathbb{B}$, that is, there exists no circle containing all points in $\mathbb{R}$ and no points in $\mathbb{B}$, Bitner *et al.* [1] present two algorithms for computing all of the smallest circles which contains all points in $\mathbb{R}$ and as few points of $\mathbb{B}$ as possible. The time complexities of these two algorithms are $O(nm \log m + n \log n)$ and $O(mn) + O^*(m^{1.5})$, respectively, where $|\mathbb{R}| = n$; $|\mathbb{B}| = m$ and $O^*()$ notation ignores polylogarithmic factors.

In this paper, given two sets of points $\mathbb{R}$ and $\mathbb{B}$, we consider the problem of finding a set of minimum number of separating circles, satisfying the condition that every point in $\mathbb{R}$ is covered by at least one circle and each point in $\mathbb{B}$ is not covered by any circle. This problem is denoted as *MNSC* for short. We conjecture that the problem is NP-hard and present an approximation algorithm for finding such a set and the lower bound of the minimum number of circles in the set.

Effective solutions find applications in wireless communication, especially when communication should not be detected by some pre-specified sites. Such requirements arise often in military applications. In such cases, the communication

978-0-7695-4483-0/11 $26.00 © 2011 IEEE
DOI 10.1109/ISVD.2011.21

should be confined within all ally sites (red points) and not be detected by any enemy site (blue point).

II. Brief Idea

Before presenting details of the algorithm, we describe the idea briefly. Given two sets of points $\mathbb{R}$ and $\mathbb{B}$, we try to find a set of circles such that every point in $\mathbb{R}$, also called red point alternatively in the following of the paper, is covered by at least one circle and each point in $\mathbb{B}$, also called blue point, is not covered by any circle. We call a circle covering no points in $\mathbb{B}$ is legal. In the special case when $\mathbb{B}$ is empty, it is straightforward that one legal circle is enough. When $\mathbb{B}$ is not empty, generally more than one legal circles are required. The Delaunay triangulation $\mathcal{DT}(\mathbb{B})$ of points in $\mathbb{B}$ is constructed. We find the circles which are not replaceable by some circle, that is, there is at least one point in $\mathbb{R}$ which can be covered only by such a circle. These circles are called *irreplaceable*. We find all irreplaceable circles. After that, we follow some greedy rule to find iteratively a Delaunay triangle whose circumcircle is taken as a separating circle to cover remaining points in $\mathbb{R}$. It is expected that removing red points covered by the circumcircle leads to more irreplaceable circles. The output is the set of all these circles found in both steps.

III. Algorithm

A. Definitions

Definition 1: Any circle containing no points in $\mathbb{B}$ is a *legal* circle.

Definition 2: Given a point p and a circle c containing p, we denote the set of red points covered by c as $\mathbb{R}_c$. If for any circle c' containing p, the set of red points covered by c' is a subset of $\mathbb{R}_c$, then c is an *irreplaceable* circle.

In Figure 1, the three dashed circles cannot contain both p and any other points outside of the solid circle. Therefore, the solid circle is an irreplaceable circle. In Figure 2, the point p locates outside of the convex hull of blue points. There are two infinite legal circles passing through p and a blue point on the convex hull, as illustrated in dashed lines. We see that neither of the two contain points outside of the solid circle. The two dashed circles cannot contain both p and any other points outside of the solid circle. We conclude that the solid circle is an irreplaceable circle. We have the following lemma relating irreplaceable circles with the set with minimum number of separating circles.

Lemma 1: Assume that $\mathbb{C} = \{c_i, i = 1, \ldots, k\}$ is a set with minimum number of separating circles (*SMNSC*) of the given two sets of points $\mathbb{R}$ and $\mathbb{B}$. For any irreplaceable circle c, there exists a circle $c_j, j \in \{1, \ldots, k\}$, such that all red points in c_j are covered by c.

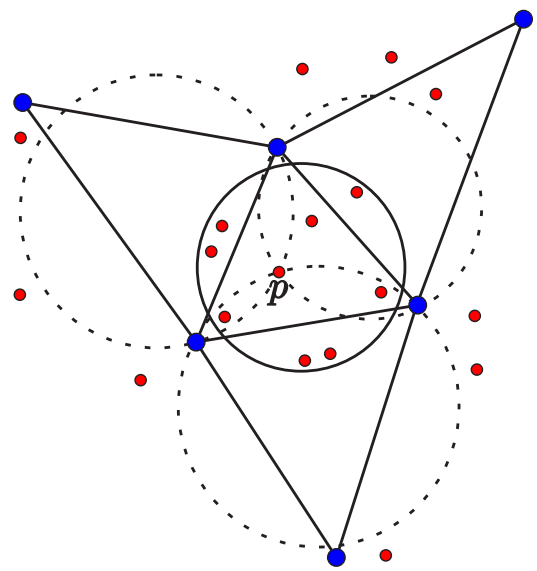


Figure 1. Illustrated in solid, an irreplaceable circle contains the point p in an interior triangle of the Delaunay triangulation of blue points (not all Delaunay triangles are shown).

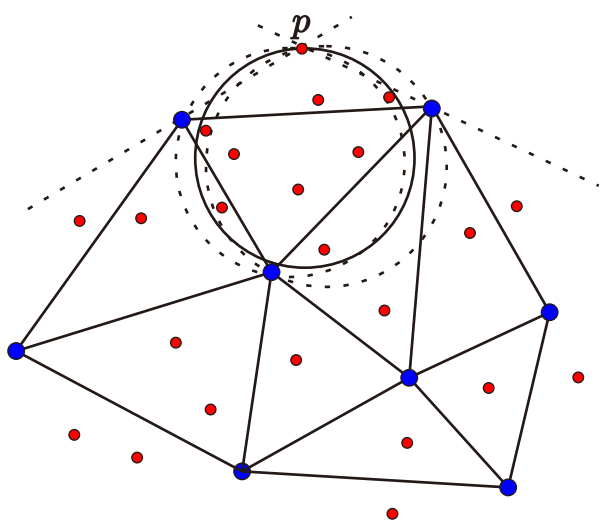
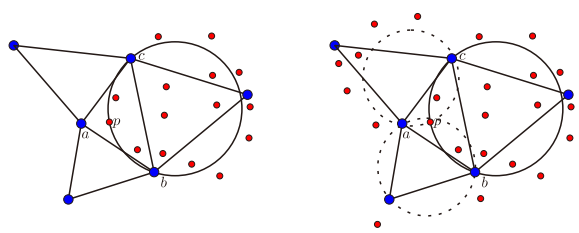


Figure 2. An irreplaceable circle contains the point p outside of the convex hull of blue points.

Proof: From definition 2, we have a red point p in the circle c such that any legal circle cannot contain both the point p and any red point outside of the circle c. In $\mathbb{C}$ there must exist a circle $\bar{c}$ containing p. According to definition 2, $\bar{c}$ does not contain any red point outside of circle c. $\bar{c}$ contains only (part of) red points covered by c. Let c_j be $\bar{c}$. This completes the proof. $\square$

From Lemma 1, the circle c_j in $\mathbb{C}$ can be replaced by c, that is, $(\mathbb{C} \setminus \{c_j\}) \cup \{c\}$ is also a SMNSC.

For a Delaunay triangle containing red points, if there is no points outside of the triangle opposite to two edges, then we can probably find an irreplaceable circle. In Figure 3 (a), the largest legal circle passing through b and c and contain all red points in the triangle abc is an irreplaceable circle. In Figure 3 (b), the solid circle is an irreplaceable circle since except for the red points in the solid circle, there are no red points inside the two dashed circles. This is the major approach of finding irreplaceable circle in the algorithm presented later in the paper, we study the effect of the distribution fo red points on an irreplaceable circle.



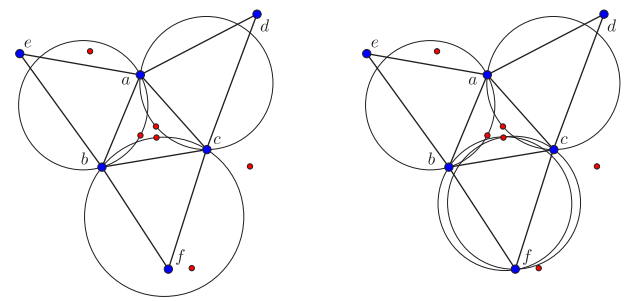
(a) (b)

Figure 3. A Delaunay triangle abc and its three adjacent triangles (not all Delaunay triangles are shown). The irreplaceable circle is in solid. (a): no legal circle can pass through a and b (or a and c) and contain any point outside of the solid circle. Loosely speaking, there are no points opposite to the edge ab (or ac) (b): no legal circle can pass through a and b (or a and c) and contain both the point p and any point outside of the solid circle, as shown by the two dashed circles.

For every triangle in $\mathcal{DT}(\mathbb{B})$, on each edge, we describe the relationship between the red points in the triangle and the red points outside of the triangle and opposite to the edge in the following two definitions. These two definitions help characterize irreplaceable circles.

Definition 3: If a triangle abc has red points inside, we define its edge *detached* (resp. *attached*) if there is no (resp. there exists a) legal circle covering some red points inside the triangle and red points outside of the triangle and opposite to the edge.

With respect to the triangle abc, the edge ac is detached with respect to the triangle abc in Figure 4 (a), since there exists no circle cover points inside the triangle abc and points outside of the triangle abc and opposite of the edge bc; the edge bc is also detached since no legal circles cover points



(a) (b)

Figure 4. The edge bc and ac are detached with respect to the triangle abc. The edge ab is attached with respect to abc.

inside the triangle abc and points outside of the triangle abc and opposite of the edge ac (ref. Fig 4); the edge ab is attached.

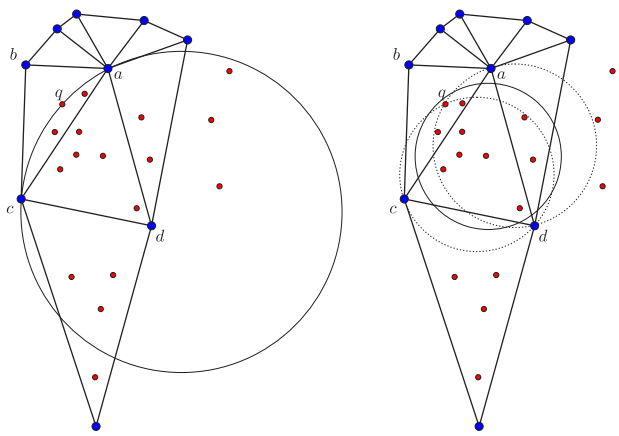
In order to test whether an edge is detached or attached with respect to a triangle, we construct the largest circle passing through the end points of the edge and one red point inside the triangle. Such circles are shown in Figure 4 (a). We can see the circle passing through points a and b (a and c) are legal; the circle passing through b and c is not legal. If this largest circle is legal and covers no red points outside of the triangle and opposite to the edge, the edge is detached. If this largest circle contains blue points, we construct two largest legal circles passing through one end point of the edge and the third point of the triangle opposite to the edge (e.g., f in Figure 4) and one red point inside the triangle, If either of the two circles contains a red point outside of the triangle and opposite to the edge, the edge is attached, otherwise, it is detached. Similarly, if one or both of the two newly constructed circles contain blue points, they will be replaced by some smaller legal circles.

After labelling detached/attached for each edge of all triangles containing red points, we have the following definition:

Definition 4: A triangle T is *d-detached*, if T has d detached edges with respect to it.

The triangle abc in Figure 4 is 2-detached, since it has two detached edges bc and ca.

If a triangle is 3-detached and has red points inside, it is obvious that the circumcircle of the triangle is an irreplaceable circle of the triangle. We may choose every red point in the triangle as the point p in definition 2. A 2-detached triangle does not always have irreplaceable circles. In Figure 5 the Delaunay triangulation of $\mathbb{B}$ is also shown. Triangle abc is a 2-detached triangle. The red points in the triangle are detached with respect to edges ab and bc. In order to find the irreplaceable circle of triangle abc, we construct a circle passing through vertices a and c, and the circle contains all the red point located in triangle abc, and one of them is



(a) (b)

Figure 5. Irreplaceable circle of a 2-detached triangle abc. (a): the dashed circle is not legal. (b): two legal circles (in dashed lines) are constructed. The solid circle is an irreplaceable circle.

on the circle (Figure 5 (a)). If the circle is legal, then it is an irreplaceable circle of triangle abc. The red point on the circle is the point p in definition 2. But the circle contains blue point d, it is not a legal circle. In order to see whether the triangle abc has an irreplaceable circle, a further test must be performed.

Draw two largest legal circles passing through the same red point q in triangle abc and blue point c and a respectively (dashed circles in Figure 5 (b)). If all the red points in the two circles and the red points in triangle abc can be covered by a legal circle (solid circles in Figure 5 (b)), this legal circle is an irreplaceable circle of triangle abc, and the point q is the point p in definition 2. Now we are sure that the triangle abc has an irreplaceable circle. In order to construct the two largest legal circles, two circles passing through point a, d, q and c, d, q are constructed respectively. If one or both of the circles have blue point in their interior, the circles are replaced by some smaller circles passing through the blue point. This processing is similar to the processing to show the red points in triangle abc are detached with respect to the edge bc in Figure 4.

If a triangle is 1-detached or 0-detached, it is possible that it has irreplaceable circles. The irreplaceable circle contains the red points in the triangle and some of points with respect to all attached edges of the triangle. From the above analysis, we know that for a triangle with large number of detached edges, it has high possibility of having irreplaceable circles.

B. Main Algorithm

Lemma 1 implies that the circle in SMNSC can be replaced by corresponding irreplaceable circle. If an irreplaceable circle can be found, one of the circles in any SMNSC can be replaced by the irreplaceable circle. We then add the circle in the current SMNSC and remove the red points in it. For the blue point set $\mathbb{B}$ and the newly formed red point set $\mathbb{R}$ with remaining red points we continue to find irreplaceable circles. This procedure terminates when all the red points are covered or there are no irreplaceable circles. If there are no red points left, all the separating circles form the set with minimum number of separating circles SMNSC. Algorithm 1 describes this approach.

Algorithm 1 Find the Set IC of Irreplaceable Circles

```
while there are still red points do
  if there exists some irreplaceable circle c then
    IC ← c;
    Remove red points covered by c;
  else
    exit Algorithm 1;
  end if
end while
```

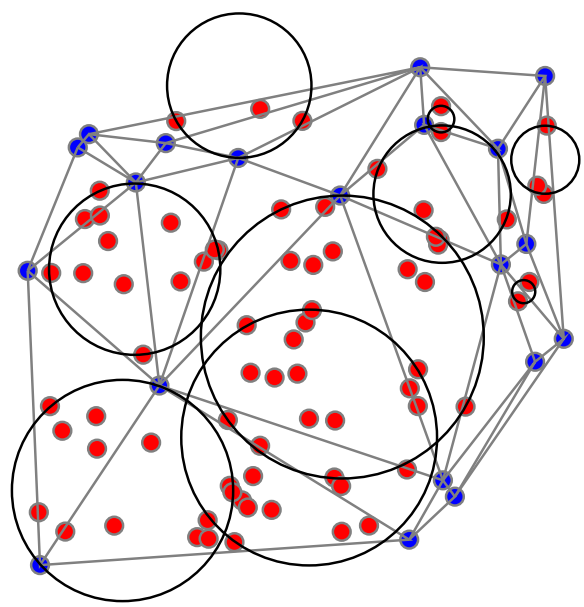


Figure 6. One example with the SMNSC found by Algorithm 1: the SMNSC has 9 separating circles, all of which are irreplaceable. The Delaunay triangulation of blue points is shown in grey.

If all red points are covered after termination of the algorithm, from Lemma 1, we obtain a SMNSC. In an example demonstrated in Figure 6, the algorithm successfully removes all red points after termination. There are 20 blue points and 74 red points in this example; the Delaunay triangulation of the blue points has 28 triangles. The SMNSC has 9 separating circles. All these circles are irreplaceable circles.

Algorithm 1 may not remove all red points. When the algorithm terminates, there may exist some triangle has red points inside and no irreplaceable circles can be found. The

following strategy is applied to solve this problem. Find a triangle with as many adjacent 1-detached triangles as possible. The circumcircle of this triangle is accepted as a separating circle, and remove all the red points in this circle. It can be proved that after the red points in the separating circle are removed, the adjacent i-detached triangles become $(i+1)$-detached triangles. The newly formed $(i+1)$-detached triangles are more likely to have irreplaceable circles than before. We have the following improved Algorithm 2.

Algorithm 2 Find the Set SC of Separating Circles

```
while there are still red points do
  if there exists some irreplaceable circle c then
    SC ← c;
    Remove red points covered by c;
  else
    if there exists a triangle containing red points then
      Find a triangle T with as many adjacent 1-
      detached triangles as possible;
      SC ← the circumcircle c of T;
      Remove red points covered by c;
    else
      exit Algorithm 2;
    end if
  end if
end while
```

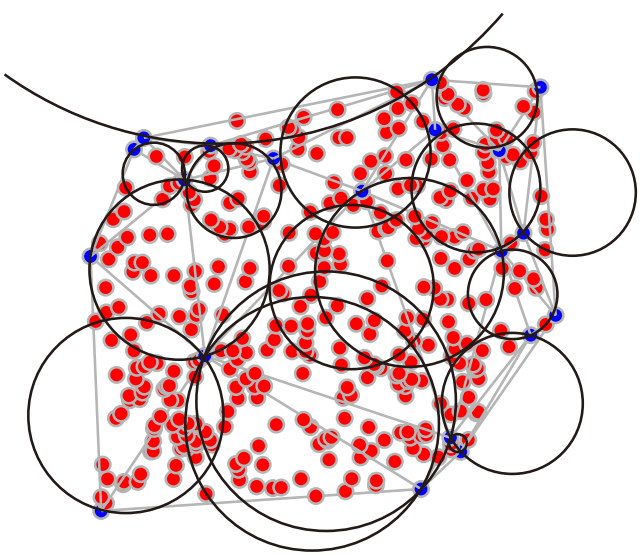


Figure 7. Another example with the SMNSC found by Algorithm 2: there are 17 separating circles, of which 12 are irreplaceable. The Delaunay triangulation of blue points is shown in grey.

In Algorithm 2 each separating circle is either an irreplaceable circle or a circumcircle of a triangle in $\mathcal{DT}(\mathbb{B})$. From Lemma 1 every irreplaceable circle generated in the algorithm can replace a circle in SMNSC, but a circumcircle generated in this algorithm may not replace a circle in SMNSC. The total number of irreplaceable circles is a lower bound of the size of SMNSC. We have the following theorem.

Theorem 1: Algorithm 2 finds the lower bound and upper bound of the size of a SMNSC.

Figure 7 demonstrates another example; the blue point set is the same as in the example shown in Figure 6. In this example there are 359 red points. The set of separating circles constructed by Algorithm 2 has 17 separating circles, 12 of which are irreplaceable circles. From Theorem 1, the lower bound and upper bound of the size of SMNSC is 12 and 17, respectively.

C. Handling Red Points outside of the Convex Hull

If there are red points outside of the convex hull of the blue points, We run Algorithm 2 first. Upon termination of Algorithm 2, the separating circles containing all the red points inside the convex hull of the blue points are found, and the red points covered by these separating circles are removed. If there exist a non-empty set $\mathbb{R}$ of remaining red points outside of the convex hull of the blue points set, the minimum number of separating circles covering all points in $\mathbb{R}$ can be found in polynomial time as shown next.

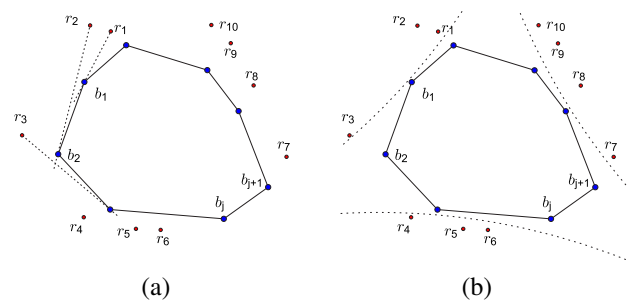


Figure 8. (a): sorted Red points. (b): The minimum number of separating circles is 3.

The vertices $\{b_j\}$ of the convex hull are ordered anticlockwise. For every red point r_i of R, a tangent r_ib_j to the convex hull of $\mathbb{R}$ is constructed, such that blue points are on the left side of the ray r_ib_j. See r_1b_1 and r_2b_2 in Figure 8 (a) for example. The points r_i of $\mathbb{R}$ are sorted by the order of the tangent point b_j of the tangent r_ib_j on the convex hull. If two red points have the same tangent point, the angle $\angle b_{j-1}r_ib_j$ are compared. The point with small angle is placed before the point with large angle. If two angles are the same, then the distance between b_j and r_i is computed. The point with small distance is placed behind. Without loss of generality, we assume the points are labelled sequentially $r_1, r_2 \ldots$ in $\mathbb{R}$.

A legal circle covering the red points of $\mathbb{R}$ on the right side of the tangent r_ib_j is named separating circle from

r_i and is denoted by $SCF(r_i)$. Now we construct a set of legal separating circles covering $\mathbb{R}$ starting from a red point $\bar{r}_i$. The set is denoted as $SLSC(\bar{r}_i)$. Assume $\bar{\mathbb{R}}$ is a sorted sequence of $\mathbb{R}$ with $\bar{r}_i$ as the first point in the sequence. Construct the separating circle $SCF(\bar{r}_i)$ from $\bar{r}_i$ such that all red points on the right side of the ray $\bar{r}_i b_j$ are inside the circle. Remove the red points covered by the circle and start from the next point left in $\bar{\mathbb{R}}$. Iteratively perform this operation until $\bar{\mathbb{R}}$ is empty. Performing this with each point r_i in $\mathbb{R}$ as the starting point will find $\|\mathbb{R}\|$ sets of separating circles. It is easy to see that the minimum number of separating circles to cover all points in $\mathbb{R}$ is one of these sets. Then, we choose the set with the least circles. Figure 8 (b) shows an example with three circles as the minimum number of separating circles, the input is the same as Figure 8 (a). The algorithm is summarized as below.

Algorithm 3 Find the Set with the Minimum Number of Separating Circles to Cover Remaining Red Points outside of the Convex Hull

```
Sort all red points in ℝ;
for each point r_i do
  R̄ = ℝ;
  while R̄ is not empty do
    Find the separating circle c to cover r_i and as many
    points behind r_i as possible;
    Add c to SLSC(r_i);
    Remove points covered by c from R̄;
  end while
end for
Output the SLSC(r_i) with the minimum number of
circles;
```

With this algorithm, we have the following lemma:

Lemma 2: The number of separating circles constructed by Algorithm 3 is minimum.

Finally we have the following theorem:

Theorem 2: If Algorithm 2 generates n separating circles, and among them m separating circles are generated circumcircles and k separating circles are found by Algorithm 3, the lower bound of the size of SMNSC is $n + k - m$ and the upper bound is $n + k$.

IV. Discussion

A. Time complexity

The Delaunay triangulation of $\mathbb{B}$ can be computed in $O(m \log m)$. The size of Delaunay triangles is $O(m)$. Find an irreplaceable circle needs $O(n)$ time in the worst case. The time to handle points in $\mathbb{R}$ outside of the convex hull of $\mathbb{B}$ is $O(nm)$ in the worst case. Therefore, we conclude that the time complexity of our algorithm is $O(m \log m + mn)$.

B. Extension to High Dimensions

Our algorithm can be extended to higher dimensions. In n-dimensional space, the concept of *detached* and *attached* is defined on $n-1$-dimensional simplex. For example, in three dimensions, these concepts are defined on triangles connecting two Delaunay tetrahedra. Irreplaceable spheres can be found similarly. However, in three or higher dimensions, the hyper-spheres covering points in $\mathbb{R}$ outside of the convex hull of $\mathbb{B}$ is not guaranteed to be found optimal since the order of red points is not defined and greedy algorithms must be adopted.

V. Conclusion and Future Work

An approximation algorithm of computing a SMNSC is proposed. After the algorithm terminates, we find the lower bound and the upper bound of the size of a SMNSC. We expect to prove that the problem is an NP-hard problem.

Acknowledgement

This research is partially supported by National Research Foundation of Singapore and National Science Foundation of China under grant No. 60573181, 61073162 and 61070093.

References

[1] Steven Bitner, Yam Cheung, and Ovidiu Daescu. Minimum separating circle for bichromatic points in the plane. In *Proceedings of the 2010 International Symposium on Voronoi Diagrams in Science and Engineering*, ISVD '10, pages 50–55. IEEE Computer Society, 2010.

[2] Jean-Daniel Boissonnat, Jurek Czyzowicz, Olivier Devillers, Jorge Urrutia, and Mariette Yvinec. Computing largest circles separating two sets of segments. In *Proceedings of the 8th Canadian Conference on Computational Geometry*, pages 173–178. Carleton University Press, 1996.

[3] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, 3rd ed. edition, 2008.

[4] H. Edelsbrunner and F. P. Preparata. Minimum polygonal separation. *Inf. Comput.*, 77:218–232, 1988.

[5] Steve Fisk. Separating point sets by circles, and the recognition of digital disks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 8(4):554 –556, 1986.

[6] Sariel Har-Peled and Soham Mazumdar. Fast algorithms for computing the smallest k-enclosing circle. *Algorithmica*, 41:147–157, 2005.

[7] Jiří Matoušek. On enclosing k points by a circle. *Inf. Process. Lett.*, 53:217–221, 1995.

[8] Nimrod Megiddo. Linear-time algorithms for linear programming in r3 and related problems. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, SFCS '82, pages 329–338. IEEE Computer Society, 1982.

[9] Nimrod Megiddo. Linear programming in linear time when the dimension is fixed. *J. ACM*, 31:114–127, 1984.

[10] Nimrod Megiddo. On the complexity of polyhedral separability. *Discrete and Computational Geometry*, 3:325–337, 1988.

[11] Joseph O'Rourke, S. Rao Kosaraju, and Nimrod Megiddo. Computing circular separability. *Discrete and Computational Geometry*, 1:105–113, 1986. 10.1007/BF02187688.

[12] Godfried T. Toussaint. Computing largest empty circles with location constraints. *International Journal of Parallel Programming*, 12:347–358, 1983.