Base station placement on boundary of a convex polygon☆

Sasanka Roy [a,1], Debabrata Bardhan [b], Sandip Das [c,*]

[a] *Tata Consultancy Services, Pune 411 013, India*
[b] *LSI Logic India Pvt. Ltd., Kolkata 700 091, India*
[c] *Indian Statistical Institute, Kolkata 700 108, India*

Abstract

Let P be a polygonal region which is forbidden for placing a base station in the context of mobile communication. Our objective is to place one base station at any point on the boundary of P and assign a range such that every point in the region is covered by that base station and the range assigned to that base station for covering the region is minimum among all such possible choices of base stations. Here we consider the forbidden region P as convex and base station can be placed on the boundary of the region. We present optimum linear time algorithm for that problem. In addition, we propose a linear time algorithm for placing a pair of base stations on a specified side of the boundary such that the range assigned to those base stations in order to cover the region is minimum among all such possible choices of a pair of base stations on that side.

Keywords: Algorithm; Wireless network; Farthest point Voronoi diagram; Computational geometry

1. Introduction

Sometimes fixing the location of the base station becomes difficult if the region is a huge water body, or a dense forest or some other prohibited zone. However, we need to provide mobile communication service inside that region. In order to minimize the power requirement (or effectively the range) of the base stations, we have to place the base stations in some appropriate locations on the boundary of that region. For the sake of simplicity, we consider that the given region is convex. Here the objective is to locate the position of base station(s) with some additional constraints such that every point inside that polygon is covered by these base station(s). In other words, every point inside that polygon is within the range of at least one base station and the maximum among the ranges of these base stations is minimized. We will consider the following two problems in the context of placing base stations on the boundary of a polygonal region.

Problem $P1$: Locate a point α on the boundary of the polygon P such that the maximum among the distances from α to all the points inside the polygon P is minimized.

Problem $P2$: Identify two points γ and δ on a given edge e of the polygon P and a real number r such that every point x inside the polygon P is covered by at least one of the circles centered at γ or δ of radius r and the value of r is minimum for such choices of γ and δ.

In Section 2, we address the problem $P1$ and propose a linear time algorithm for computing the location α on the boundary of the polygon P. In Section 3, a linear time algorithm for problem $P2$ is proposed.

2. Problem $P1$

Euclidean 1-center problem is a well-known problem which has a long history. Here the problem is to find the smallest circle that encloses a given set of n points. In the standard 1-center problem, there is no restriction on placement of the center of that circle. Shamos and Hoey [13], and Preparata [12] initially proposed different $O(n \log n)$ time algorithms which are a considerable improvement over $O(n^2)$ solution proposed in [7]. Lee [10] proposed the farthest point Voronoi diagram structure and using that structure, the 1-center problem can be solved in $O(n \log n)$ time. Finally Megiddo [11] found an optimal $O(n)$ time algorithm for solving this problem using prune-and-search technique.

While much has been done on such unconstrained version of the classical problem, little has been done in constrained version. Some interesting results were provided by Megiddo [11] and Hurtado et al. [8]. Megiddo in [11] studied the case where the center of the smallest enclosing circle must lie on a given straight line. Hurtado et al. [8] used linear programming to provide an $O(n+m)$ time algorithm for finding minimum enclosing circle with its center satisfying m linear constraints.

We will address problem $P1$ with a different type of constraint. Instead of placing the center inside a given convex region, we consider the center on the boundary of the given convex polygon, and the objective is to cover the entire region inside the polygon. A similar problem was first addressed by Bose and Toussaint [5], where the center of the minimum enclosing circle lies on the boundary of a convex polygon of size n and the objective is to cover a set of m points which may not lie on or inside the polygon. An $O((n+m)\log(n+m))$ time algorithm for that problem was also presented in that paper. Here we derive some interesting geometric characterizations and propose an $O(n)$ time algorithm for problem $P1$ that avoids the use of linear programming techniques.

Let the vertices of the convex polygon P be $\{v_0, v_1, \ldots, v_{n-1}\}$ in anticlockwise order. We will use e_i to denote the edge (v_i, v_{i+1}) of P. If $\mathscr{C}$ denotes the minimum radius circle enclosing P whose center α is on the boundary of P, then $\mathscr{C}$ must satisfy the following simple but interesting observations.

Observation 1. *The circle $\mathscr{C}$ must pass through at least one vertex of the polygon P.*

Observation 2. *Let e be the edge of the polygon P that contains the point α. If the circle $\mathscr{C}$ passes through exactly one vertex v of polygon P, then the line $\overline{v\alpha}$ is perpendicular to the edge e at point α.*

Let us consider the farthest point Voronoi diagram $\mathscr{V}(P)$ of the vertices $\{v_0, v_1, \ldots v_{n-1}\}$ of the polygon P. It partitions the plane into regions, $\mathscr{R}(v_0), \mathscr{R}(v_1), \mathscr{R}(v_2), \ldots, \mathscr{R}(v_{n-1})$, such that for any point $p \in \mathscr{R}(v_j)$, $d(p, v_j) \geqslant d(p, v_i)$ for all $v_i \in P$, where $d(.,.)$ denotes the Euclidean distance between a pair of points. From Observation 1, we can conclude that if v_i is on boundary of $\mathscr{C}$ then v_i is farthest vertex from α and hence α must be in $\mathscr{R}(v_i)$. Sometimes the circle $\mathscr{C}$ may pass through more than one vertex of the polygon P, and in that case we have the following observation.

Observation 3. *If the circle $\mathscr{C}$ passes through two vertices of polygon P, then α must be at the intersection point of an edge of $\mathscr{V}(P)$ with an edge of the polygon P. Moreover, if $\mathscr{C}$ is passing through more than two vertices of polygon P, then α coincides with a vertex of $\mathscr{V}(P)$.*

From the above observations, we can conclude the following lemma.

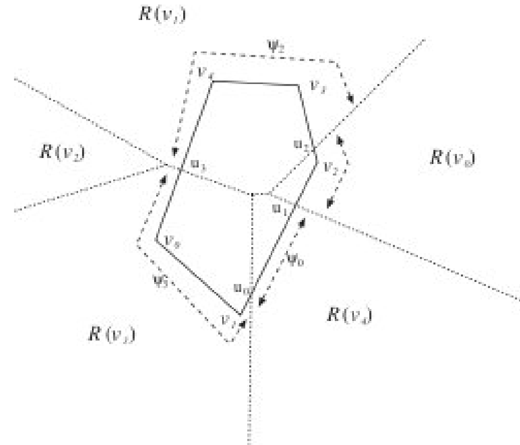

Fig. 1. Illustrating the proof of Lemma 2.

Lemma 1. *If the center α of the circle $\mathscr{C}$ lies on an edge e of the polygon P, then α must coincide with either the perpendicular projection of some vertex of P on the edge e, or with the intersection point of an edge of $\mathscr{V}(P)$ and the edge e of P.*

Proof. Lemma follows from Observations 2 and 3. □

We consider each edge e of P, and locate the vertices of P whose projection on e lies inside the edge segment e. We use $\mathscr{B}$ to denote the set of these points on e which are obtained by these projections. Note that, if a point $\beta \in \mathscr{B}$ is the projection of a vertex v on an edge of P and $\beta \in \mathscr{R}(v)$, then the smallest enclosing circle of P with center at β passes through v. So, we consider only those members in $\mathscr{B}$ that lie in the farthest point Voronoi region of the corresponding vertex of P. Each element $\beta \in \mathscr{B}$ is attached with a vertex v such that $\beta \in \mathscr{R}(v)$. In Fig. 1, the projection of the vertex v_1 on the edge (v_3, v_4) is the only member in $\mathscr{B}$ that is being considered.

In order to compute the smallest enclosing circle with center on the boundary of P and which passes through two vertices of P, we need to compute the set of points $\mathscr{A}$ that are generated due to the intersections of the edges of $\mathscr{V}(P)$ with the edges of the polygon P. Each element $u \in \mathscr{A}$ is attached with a pair of vertices (v, v') of P such that u is the point of intersection of the boundary of P and the Voronoi edge which correspond to the bisector of the line segment $[v, v']$ and v' appears after v along the boundary of P in anticlockwise order. The members in set $\mathscr{A}$ partition the boundary of polygon into a set of polygonal chains. Each of these chains must lie inside a single Voronoi region, and it is formed by a sequence of polygonal edges bounded by a pair of consecutive members of set $\mathscr{A}$. Let us denote the set of these polygonal chains as $\mathscr{D}(P)$. In Fig. 1, $u_0, u_1, \ldots, u_3$ are the elements of $\mathscr{A}$ and the chains $\psi_0, \psi_1, \ldots, \psi_3$ are in $\mathscr{D}(P)$.

The computation of the sets $\mathscr{A}$ and $\mathscr{B}$ using the farthest point Voronoi diagram needs $O(n \log n)$ time. Although the farthest point Voronoi diagram for the vertices of a convex polygon can be computed in linear time [1], computation of all the intersection points of the Voronoi edges with polygonal boundary of polygon P needs $O(n \log n)$ time (as stated by Bose and Toussaint [5]).

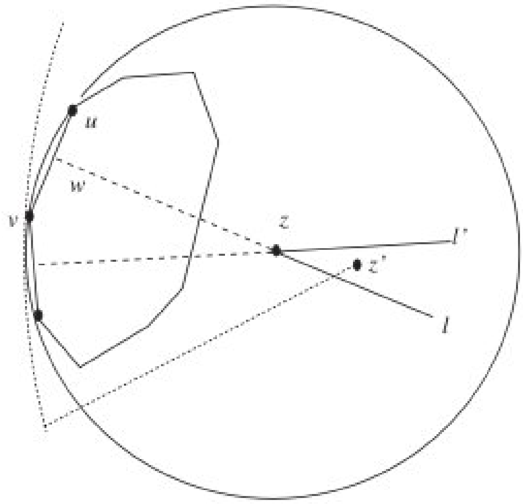

Fig. 2. Illustrating the proof of Lemma 3.

We present a simple $O(n)$ time algorithm for finding the set $\mathscr{A}$ and $\mathscr{B}$. This avoids the computation of the farthest point Voronoi diagram. But, we need to study the properties of the farthest point Voronoi diagram of the vertices of a convex polygon, which will help in the formulation of our algorithm.

Lemma 2. *Each cell of $\mathscr{V}(P)$ is an unbounded convex region.*

Proof. Let b be the perpendicular bisector of line segment $[v_j, v_k]$ that separates $\mathscr{R}(v_j)$ and $\mathscr{R}(v_k)$. Note that, $\mathscr{R}(v_j)$ must lie in the side opposite to v_j with respect to the line b. Similarly it can be shown that, v_j lies in the intersection of all the half planes defined by the lines on the boundary of $\mathscr{R}(v_j)$. Thus, $\mathscr{R}(v_j)$ is an unbounded cell. Again, $\mathscr{R}(v_j)$ is convex due to the fact that it is the intersection region of a set of half-planes. □

Lemma 3. *Let e be an edge of the polygon P. The perpendicular bisector of e must define one of the edges of the boundary of a Voronoi cell of $\mathscr{V}(P)$. Furthermore, this boundary is a half-line.*

Proof. Let l be the perpendicular bisector of an edge $e = (u, v)$ of P. The line l intersects e at a point w. If we move along the line l from w on the direction toward the interior of the polygon then we can locate a point z on line l such that the circle centered at z with radius equal to $d(z, u)$ encloses all other vertices of P. Hence, the portion of l from z toward the other side of w is a half line, and it defines Voronoi edge separating $\mathscr{R}(u)$ and $\mathscr{R}(v)$ (see Fig. 2). □

Lemma 4. *If the perpendicular bisectors of a pair of adjacent edges $e_i = (v_{i-1}, v_i)$ and $e_{i+1} = (v_i, v_{i+1})$ of a convex polygon P intersect outside P, then$P \cap \mathscr{R}(v_i) = \emptyset$.*

Proof. Let l_1 and l_2 be the perpendicular bisectors of e_i and e_{i+1} respectively and they intersect at a point q outside P. The Voronoi cells $\mathscr{R}(v_{i-1})$ and $\mathscr{R}(v_i)$ are in the two different half

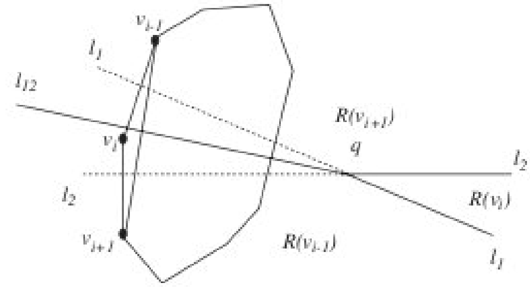

Fig. 3. Illustrating the proof of Lemma 4.

planes defined by the line l_1. Similarly, the Voronoi cells $\mathscr{R}(v_i)$ and $\mathscr{R}(v_{i+1})$ are in the two different half planes defined by the line l_2. The Voronoi cell $\mathscr{R}(v_i)$ is in the common region of the aforesaid two half planes as shown in Fig. 3, which is outside the polygon P. □

From the above lemma, we can conclude that the intersection of a Voronoi cell with the boundary of P is a simple contiguous chain and hence the cardinality of the set $\mathscr{A}$ is less than or equal to the number of vertices in P. Let $u_0, u_1, \ldots, u_{k-1}$ be the points in set $\mathscr{A}$ and they are in anticlockwise order on the boundary of the polygon P. As mentioned earlier, the members in $\mathscr{A}$ define the set of polygonal chains $\mathscr{D}(P)$. Let $\mathscr{D}(P) = \{\psi_0, \psi_1, \ldots, \psi_{k-1}\}$, where the chain ψ_i is bounded by the points u_i and u_{i+1}. Let $\psi_i \in \mathscr{R}(v'_i)$, where $v'_i \in \{v_0, v_1, \ldots, v_{n-1}\}$. The farthest neighbor of all the vertices in ψ_i (if exists) is v'_i. We will use $f(v)$ to denote the farthest neighbor of vertex v, $v \in P$. If $v \in \psi_i$, then $f(v) = v'_i$. We would also introduce a new function $index()$, where $index(v'_i) = j$ whenever $v'_i = v_j$. Here in Fig. 1, $index(v'_0)$, $index(v'_1)$, $index(v'_2)$ and $index(v'_3)$ are 4, 0, 1 and 3, respectively. The following lemma demonstrates the arrangement of Voronoi cells along the boundary of the polygon.

Lemma 5. *If $index(v'_r) = \min\{index(v'_0), index(v'_1), \ldots, index(v'_{k-1})\}$, then $index(v'_r) < index(v'_{r+1}) < \cdots < index(v'_{k-1}) < index(v'_0) < \cdots < index(v'_{r-1})$.*

Proof. Two adjacent polygonal chains ψ_r and ψ_{r+1} meet at point u_{r+1}, which is on the perpendicular bisector (say l) of the line segment joining the vertices $v'_r, v'_{r+1} \in P$. The vertex v'_r (resp. v'_{r+1}) and the polygonal chain ψ_r (resp. ψ_{r+1}) lie in different sides of the line l as shown in Fig. 4. So, a circle C centered at u_{r+1} with radius $d(u_{r+1}, v'_r)$ passes through v'_r and v'_{r+1}, and contains the polygon P. As the vertex v'_{r+1} is on anticlockwise direction of v'_r, and the index of v'_r is the least among all the indices of v'_i $(0 \leqslant i < k)$, we have $index(v'_r) < index(v'_{r+1})$.

Next, we prove the remaining part of the result. Note that, any circle with center on the boundary of P and containing P does not intersect the circular arc $\widetilde{v'_r v'_{r+1}}$ where $\widetilde{v'_r v'_{r+1}}$ denotes the arc from v'_r to v'_{r+1} on C in anticlockwise direction (see Fig. 4). If $index(v'_{r+1}) - index(v'_r) > 1$, then for any integer $\beta \in [index(v'_r), index(v'_{r+1})]$, there does not exist a point on

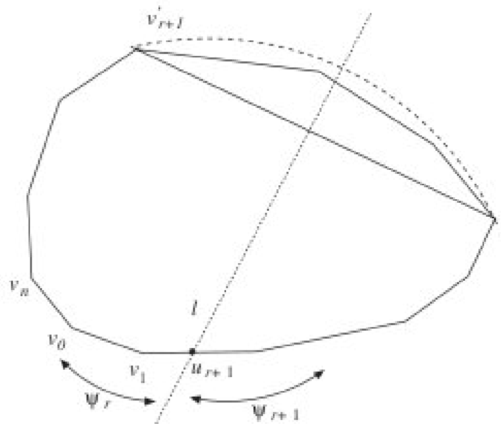

Fig. 4. Illustrating the proof of Lemma 5.

boundary of polygon P from which v_β is farthest among all vertices of polygon P. Therefore, $index(v'_j) > index(v'_{r+1})$ for all $j = 0, 1, \ldots, r-1, r+2, \ldots, k-1$. Hence, the lemma follows. □

2.1. Algorithm

We first compute two arrays $\mathscr{A}$ and $\mathscr{B}$. One of the elements from $\mathscr{A}$ or $\mathscr{B}$ will be the center of the maximum enclosing circle of P.

Algorithm. **Computation-of-Array-$\mathscr{A}$**

Input: Polygon P with a set of n vertices.

Output: The array $\mathscr{A}$ generated in anticlockwise order.

Procedure: We traverse the vertices of P in anticlockwise order. If for two consecutive vertices v_i and v_{i+1}, $f(v_i) \neq f(v_{i+1})$, then we compute the u_is' (the members in $\mathscr{A}$) that lie on edge $e_i = (v_i, v_{i+1})$ as follows:

Let $v_\gamma = f(v_i)$ and $v_\delta = f(v_{i+1})$.
(* For every pair of vertices v_j and $v_{j'}$ with $\gamma \leqslant j < j' \leqslant \delta$, their perpendicular bisector intersects e_i *)
Set $j = \gamma$ and $j' = j + 1$.
Repeat the following steps *until* $j' = \delta$

Step 1: Draw the perpendicular bisector of v_j and $v_{j'}$, and compute its intersection point u.

Step 2: *If* u appears to the right (toward anticlockwise direction) of the last element of array $\mathscr{A}$,
then add u in the array $\mathscr{A}$ with the pair of vertices $(v_j, v_{j'})$.
Set $j = j'$ and $j' = j' + 1$.

Step 3: *If* u appears to the left (toward clockwise direction) of the last element of array $\mathscr{A}$, *then*
(* the region $\mathscr{R}(v_j)$ does not intersect the boundary of P*)
delete the last element of $u' \in \mathscr{A}$.
If the pair of vertices attached to u' is $(v_\theta, v_{\theta'})$ where $\theta > \theta'$, then set $j = \theta'$
(* j will never go beyond γ, because $\mathscr{R}(v_\gamma)$ intersects e_i*).

Algorithm. **Computation-of-Array-$\mathscr{B}$**

Input: The polygon P and the array $\mathscr{A}$.

Output: The array $\mathscr{B}$ generated in anticlockwise order.

Procedure: Traverse the array $\mathscr{A}$ to extract the chains $\{\psi_i, i = 1, 2, \ldots, k\}$. Each ψ_i is attached with the corresponding v'_i.
For each $i = 1, 2, \ldots, k$ *do*
For each edge/edge-segment $e \in \psi_i$ *do*
Draw the perpendicular projection of v'_i on e. Let it be w.
If w lies inside e, *then* add it in $\mathscr{B}$ with the vertex v'_i.

Lemma 6. *The elements of set $\mathscr{A}$ and $\mathscr{B}$ can be located in $O(n)$ time.*

Proof. We use the monotone matrix searching technique to compute the farthest neighbor $f(v_i)$ for every vertex v_i of the convex polygon P in $O(n)$ time [3]. Note that, the vertex v_i lies in Voronoi cell corresponding to vertex $f(v_i)$ and the vertices $f(v_0), f(v_1), \ldots, f(v_{n-1})$ are in anticlockwise order, and the chains $\psi_0, \psi_1, \ldots, \psi_{k-1}$ are also in anticlockwise order. There may exist some Voronoi cell $\mathscr{R}(v)$ that does not contain any vertex of P. If in addition, $P \cap \mathscr{R}(v) \neq \emptyset$, then the chain in the cell $\mathscr{R}(v)$ is a segment of an edge of P, and that segment is bounded by two consecutive members in $\mathscr{A}$. If $v_\gamma = f(v_i)$ and $v_\delta = f(v_{i+1})$, then $P \cap \mathscr{R}(v_j)$ is either empty or a segment of edge e_i for each $j = \gamma + 1, \ldots, \delta - 1$. Each boundary of those segments is defined by the point of intersection between the edge e_i and the perpendicular bisector of the joining line segment of two vertices say, $[v_{j_1}, v_{j_2}]$, where $\gamma \leqslant j_1 < j_2 \leqslant \delta$. Those perpendicular bisectors are the boundaries of the Voronoi cells $\mathscr{R}(v_\gamma), \mathscr{R}(v_{\gamma+1}), \ldots, \mathscr{R}(v_\delta)$. We identify these members of $\mathscr{A}$ (on e_i) by observing the boundary of the Voronoi cell corresponding to vertices $v_\gamma, v_{\gamma+1}, \ldots, v_{\delta-1}$ in that order. Step 3 indicates the case where the perpendicular bisectors of segments $[v_\theta, v_j]$ and $[v_j, v_{j'}]$ intersect outside polygon P and hence from Lemma 4, $P \cap \mathscr{R}(v_j)$ is empty and therefore the newly enumerated u appears to the left of the last element of array $\mathscr{A}$. Note that, in case $\mathscr{R}(v_\theta) \cap e_i \neq \emptyset$, $\mathscr{R}(v_{j'}) \cap e_i \neq \emptyset$ and $\mathscr{R}(v_j) \cap e_i = \emptyset$ for $j = \theta + 1, \theta + 2, \ldots, j' - 1$, then the separator u ($\in e_i$), separating segments $\mathscr{R}(v_\theta) \cap e_i$ and $\mathscr{R}(v_{j'}) \cap e_i$, is the intersection point of e_i with the perpendicular bisector of segment $[v_\theta, v_{j'}]$. We execute Steps 1–3 for each of the n vertices of P, and each iteration produces an u in $\mathscr{A}$. In addition, for the deletion of each element in $\mathscr{A}$ (in Step 3) we execute Steps 1–3 once. If the final length of $\mathscr{A}$ is k, $n - k$ elements of $\mathscr{A}$ will be deleted. Thus, the result follows. □

Now, we are in position to present the algorithm for identifying optimum location of the center of the minimum enclosing circle of P on boundary of the P.

Algorithm Problem_$P1(P)$

Input: Polygon P with vertex set of n vertices.

Output: The point α on boundary of P.

Step 1: Compute the set $\mathscr{A} = \{u_0, u_1, \ldots\}$ using algorithm *Computation-of-Array-$\mathscr{A}$*.
Assign $d_{min} = \infty$.
For each $u_i \in \mathscr{A}$ *do*
(* Let v_i' be the one of the farthest neighbors of u_i, and is attached to u_i *)
If $d(u_i, v_i') < d_{min}$, *then* assign $\alpha = u_i$ and $d_{min} = d(u_i, v_i')$.

Step 2: Compute the set $\mathscr{B} = \{w_0, w_1, \ldots\}$ using algorithm *Computation-of-Array-$\mathscr{B}$*.
For each $w_i \in \mathscr{B}$ *do*
(* Let v_j be the vertex attached to w_i *)
If $d(w_i, v_j) < d_{min}$, *then* set $d_{min} = d(w_i, v_j)$, and set $\alpha = w_i$.

We now have the following theorem stating the time complexity of our proposed algorithm for problem P1.

Theorem 1. *Algorithm Problem_$P1(P)$ computes the location α in $O(n)$ time.*

3. Problem $P2$

Here we consider other variation of this problem. Instead of placing one center on the boundary, we place two base stations in order to cover the region. Given a set S, of n points, the *2-center problem* for S is to cover S by two closed disks whose radii are as small as possible. In [14], Sharir presents a near-linear time algorithm running in $O(n \log^9 n)$ time. Currently best algorithm for its solution is proposed by Chan [6]. He suggested a deterministic algorithm that runs in $O(n \log^2 n(\log\log n)^2)$ and a randomized algorithm that runs in $O(n \log^2 n)$ time with high probability. A variation of this problem is the *discrete two-center problem* that finds two closed disks whose union cover the point set S and whose centers are at points of S. This problem is solved in $O(n^{4/3} \log^5 n)$ time by Agarwal et al. [4]. Recently, Kim et al. [9] solve both of the standard and discrete two-center problem for a set of points that are in convex positions in $O(n \log^3 n \log\log n)$ and $O(n \log^2 n)$ time, respectively. A detail review about this problem can be found in the paper by Shin et al. [15].

In this section, we consider a different set of constraints while placing two circles for covering the convex polygonal region P. We consider the case, where only one edge of P is available for placing base stations. The problem in general form is hard to solve. The problem of computing a single constrained circle $\mathscr{C}$ of minimum radius which covers P, can be computed in $O(n)$ time by considering the ψ_is' which share the edge e. The procedure is very much similar to that of Problem P1. Our objective to compute two constrained equal circles for covering the entire region P with minimum radius. We propose an $O(n)$ time algorithm for this problem.

Without loss of generality, we may assume that the given edge e lies on the x-axis, it joins the vertices v_0 and v_1, and $x(v_0) < x(v_1)$. We use the term *constrained circle* to denote a circle whose center lies on e. Let $\mathscr{C}_1$ and $\mathscr{C}_2$ be two constrained equal circles of minimum radius that cover P. Let α and β be the centers of $\mathscr{C}_1$ and $\mathscr{C}_2$ respectively, and $x(\alpha) < x(\beta)$. Now, we have the following simple observations.

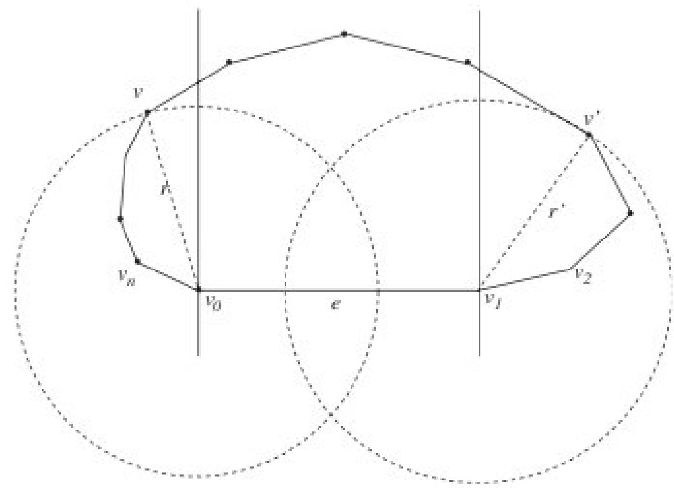

Fig. 5. Illustrating the proof of Observation 4.

Observation 4. *If v is the vertex of P such that $d(v, v_0) = \text{Max}\{d(v_i, v_0) | x(v_i) \leqslant x(v_0)\}$, and v' is the vertex such that $d(v', v_1) = \text{Max}\{d(v_i, v_1) | x(v_i) \geqslant x(v_1)\}$, then the radius of each circle $\mathscr{C}_1$ and $\mathscr{C}_2$ is greater than or equal to $\max(d(v, v_0), d(v', v_1))$ (see Fig. 5).*

Observation 5. *If a vertex v is inside $\mathscr{C}_1$ but not inside $\mathscr{C}_2$ and a vertex v' is inside $\mathscr{C}_2$ but not inside $\mathscr{C}_1$, then $x(v) < x(v')$.*

Note that, the radius of $\mathscr{C}$ (the constrained circle of minimum radius enclosing P) is greater than or equal to the maximum among the radii of $\mathscr{C}_1$ and $\mathscr{C}_2$. If $\mathscr{C}$ passes through only one vertex of P, then its radius is exactly equal to the maximum radius among the circles $\mathscr{C}_1$ and $\mathscr{C}_2$. This situation can be handled in linear time as mentioned for the problem P1. From now onwards, we assume that the circles $\mathscr{C}_1$ and $\mathscr{C}_2$ are smaller than the circle $\mathscr{C}$.

Let $v_k = \max_i\{v_i \notin \mathscr{C}_1\}$ and $v_{k'} = \min_i\{v_i \notin \mathscr{C}_2\}$. The vertices $\{v_0, v_{n-1}, v_{n-2}, \ldots, v_{k+1}\}$ are all in $\mathscr{C}_1$ and the vertices $\{v_1, v_2, \ldots, v_{k'-1}\}$ are all in $\mathscr{C}_2$. From Observation 5, we can conclude that $x(v_{k'}) < x(v_k)$. We first compute two constrained circles which cover the vertices $\{v_0, v_{n-1}, v_{n-2}, \ldots, v_{k+1}\}$ and $\{v_1, v_2, \ldots, v_{k'-1}\}$ respectively. These help in computing $\mathscr{C}_1$ and $\mathscr{C}_2$.

3.1. Algorithm

We will use $P_1(s)$ to denote the convex polygon with vertices $\{v_0, v_{n-1}, v_{n-2}, \ldots, v_s\}$, where s may assume values $0, n-1, n-2, \ldots, 1$. Similarly, $P_2(s)$ denotes the convex polygon with vertices $\{v_1, v_2, \ldots, v_s\}$, where s may assume values $1, 2, \ldots, n-2, n-1, 0$.

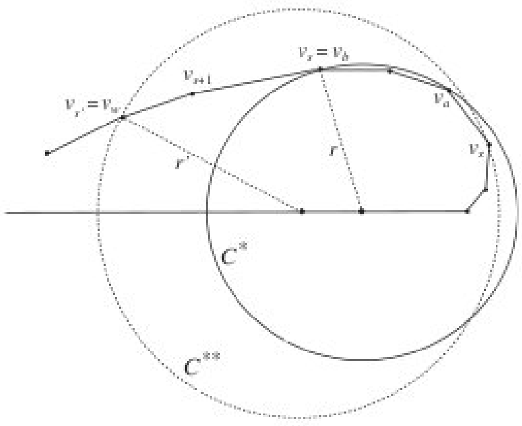

Fig. 6. Illustrating the proof of Lemma 7.

Lemma 7. *Let $\mathscr{C}^*$ and $\mathscr{C}^{**}$ be two minimum radius constrained circles enclosing the vertices of $P_2(s)$ and $P_2(s')$ respectively.*

1. *If $s' > s$ then the radius of $\mathscr{C}^*$ is less than or equal to the radius of $\mathscr{C}^{**}$.*
2. *If $s' > s$ then the x-coordinate of the center of $\mathscr{C}^*$ is greater than or equal to the x-coordinate of the center of $\mathscr{C}^{**}$.*
3. *Suppose both the circles pass through exactly two vertices of the polygon P and $s' > s$. If $\mathscr{C}^*$ passes through the vertices v_a and v_b with $a < b$, and $\mathscr{C}^{**}$ passes through the vertices v_z and v_w with $z < w$, then $z \leqslant a < b \leqslant w$.*

Proof. Suppose the radius of the circle $\mathscr{C}^*$ is r and it is centered at a point p on edge e. Since $\mathscr{C}^*$ encloses up to the vertex v_s, and p is on the left of v_1, then $x(v_s) \leqslant x(p)$, and also $x(v_i) < x(p)$ for all $i = s+1, s+2, \ldots, s'$. Since $\mathscr{C}^{**}$ encloses the points in $P_2(s)$ and also the points $v_{s+1}, v_{s+2}, \ldots, v_{s'}$, we have r less than or equal to the radius of $\mathscr{C}^{**}$ (see Fig. 6). If r is strictly less than the radius of $\mathscr{C}^{**}$, then at least one point v_i ($\in \{v_{s+1}, v_{s+2}, \ldots, v_{s'}\}$) is outside $\mathscr{C}^*$ and $\max_{i=s+1}^{s'} d(p, v_i)$ is greater than or equal to the radius of $\mathscr{C}^{**}$. Hence follows the second statement of the lemma.

Observe that, $\mathscr{C}^{**}$ does not properly contain $\mathscr{C}^*$, otherwise we can reduce the size of the constrained circle covering all the points of $P_2(s')$ by moving the center of $\mathscr{C}^{**}$ a small amount toward left. Using similar arguments, we can say that the arc $\widetilde{v_a v_b}$ of $\mathscr{C}^*$ must be inside $\mathscr{C}^{**}$ where $\widetilde{v_a v_b}$ denotes the arc from v_a to v_b in anticlockwise direction. Hence the lemma follows. □

Lemma 7 says that, if the minimum radius constrained circle $\mathscr{C}'$ covers the vertices of $P_1(s)$, and $s \geqslant k+1$, then its center is either at α or at the left side of α, where α is the center of $\mathscr{C}_1$. Similarly, for any $s' \leqslant k'-1$, if the minimum radius constrained circle $\mathscr{C}''$ encloses the vertices of $P_2(s')$, then its center is either at β or at the right side of β, where β is the center of $\mathscr{C}_2$.

Again from Lemma 1, we can conclude that the center of circle $\mathscr{C}''$ is either on edge e with x-coordinate $x(v_i)$, $1 \leqslant i \leqslant s'$ or it is at the intersection point of e with an edge of the farthest point Voronoi diagram of the vertices $\{v_1, v_2, \ldots v_{s'}\}$ of $P_2(s')$. Let $\mathscr{A}_2(s') = \{u_0, u_1, u_2, \ldots, u_m\}$ be the set of intersection points of the farthest point Voronoi diagram of $P_2(s')$ with the edge e. With each element of $\mathscr{A}_2(s')$, the corresponding pair of vertices is attached as in problem P1. The center of the desired circle $\mathscr{C}''$ to cover the vertices of $P_2(s')$ is either the perpendicular projection of $v_{s'}$ on e or one of the members in $\mathscr{A}_2(s')$ for which the radius is minimum. Similarly $\mathscr{A}_1(s)$ is also computed for the polygon $P_1(s)$. Therefore we are interested about the intersection points of e with the edges of two farthest point Voronoi diagrams with the vertices of $P_1(s)$ and $P_2(s')$ respectively.

Initially while preprocessing the point set, we do not have any prior information of k and k' that determines $\mathscr{C}_1$ and $\mathscr{C}_2$. Initially, we start with $P_1(n-1) = \{v_0, v_{n-1}\}$. Given $\mathscr{A}_1(l)$ for the polygon $P_1(l)$, we incrementally compute $\mathscr{A}_1(l-1)$ for the polygon $P_1(l-1)$ by adding the next vertex v_{l-1}. The same procedure is followed for computing $\mathscr{A}_2(l)$ for all $l = 2, 3, \ldots, n-1, 0$. The following lemma describes an important relationship between $\mathscr{A}_2(l)$ and $\mathscr{A}_2(l+1)$, computed for $P_2(l)$ and $P_2(l+1)$ respectively.

Lemma 8. *Let $\mathscr{A}_2(l) = \{u_0, u_1, u_2, \ldots, u_m\}$, and $x(u_0) \leqslant x(u_1) \leqslant x(u_2) \leqslant \cdots \leqslant x(u_m)$. After introducing the next vertex v_{l+1}, $\mathscr{A}_2(l+1) = \{u'_0, u'_1, u'_2, \ldots, u'_t\}$, and $x(u'_0) \leqslant x(u'_1) \leqslant x(u'_2) \leqslant \cdots \leqslant x(u'_t)$. Then,*

(i) $m+1 \geqslant t$,
(ii) $u_0 = u'_0, u_1 = u'_1, \ldots, u_{t-1} = u'_{t-1}$.

Furthermore, if u_{t-1} is generated by the perpendicular bisector of the line segment $[v_i, v_{i+j}]$ $(i, j > 0)$, then the point u'_t is generated due to intersection of edge e and the perpendicular bisector of line segment $[v_{i+j}, v_{l+1}]$.

Proof. Observe that the vertex v_l is farthest from any point in the segment $[u_m, v_1]$ on the edge e among the vertices $\{v_1, v_2, \ldots, v_l\}$. If the vertex v_{l+1} is not included in the circle centered at u_m passing through v_l, then there exist a region on e such that, from any point on that region, the distance of v_{l+1} is more than that of v_l. In that case, after including the vertex v_{l+1}, it may happen that the vertex v_{l+1} is farthest from any point in the segment $[x(u_m), x(v_1)]$ and hence, $e \cap \mathscr{R}(v_l) = \emptyset$, where $\mathscr{R}(v)$ represents the Voronoi cell for vertex v in the farthest point Voronoi diagram of $\{v_1, v_2, \ldots, v_{l+1}\}$. Hence from Lemmas 4 and 5, we can conclude that, there exist a vertex v_s $(1 \leqslant s \leqslant l)$ such that, $e \cap \mathscr{R}(v_s) \neq \emptyset$ and $e \cap \mathscr{R}(v_\alpha) = \emptyset$ for all α, $s < \alpha \leqslant l$ and u'_t is the point of intersection between the edge e and the perpendicular bisector of line segment $[v_{l+1}, v_s]$. As u'_{t-1} is generated by the perpendicular bisector of the line segment $[v_{i+j}, v_i]$, therefore, $s = i + j$. Note that, if v_{u_λ} is the farthest vertex among $\{v_1, v_2, \ldots, v_l\}$ from any point on segment $[u_\lambda, u_{\lambda+1}]$ for $1 \leqslant \lambda \leqslant t-1$, then v_{u_λ} is also the farthest vertex among $\{v_1, v_2, \ldots, v_{l+1}\}$ from any point on segment $[u_\lambda, u_{\lambda+1}]$. □

The same incremental result for $\mathscr{A}_1(l)$ is as follows:

Lemma 9. *Let $\mathscr{A}_1(l) = \{u_0, u_1, u_2, \ldots, u_{m'}\}$, and $x(u_0) \geqslant x(u_1) \geqslant x(u_2) \geqslant \ldots \geqslant x(u_{m'})$. After introducing the next vertex*

v_{l-1}, $\mathscr{A}_1(l-1) = \{u'_0, u'_1, u'_2, \ldots, u'_{t'}\}$, *and* $x(u'_0) \geqslant x(u'_1) \geqslant x(u'_2) \geqslant \ldots \geqslant x(u'_{t'})$. *Then,*

(i) $m' + 1 \geqslant t'$,
(ii) $u_0 = u'_0, u_1 = u'_1, \ldots, u_{t'-1} = u'_{t'-1}$.

Furthermore, if $u_{t'-1}$ *is generated by the perpendicular bisector of the line segment* $[v_i, v_{i-j}]$ $(i, j > 0)$, *then the point* $u'_{t'}$ *is generated due to intersection of edge e and the perpendicular bisector of line segment* $[v_{i-j}, v_{l-1}]$.

Lemma 10. *The total time complexity for computing* $\{\mathscr{A}_1(l), l = n-1, n-2, \ldots, 1\}$ *is* $O(n)$ *time. Another pass of same time complexity computes* $\{\mathscr{A}_2(l), l = 2, 3, \ldots, n-1, 0\}$.

Proof. We compute $\mathscr{A}_1(n-1), \mathscr{A}_1(n-2), \ldots, \mathscr{A}_1(1)$ iteratively considering n vertices of P one at a time. Each iteration we delete a sequence of u's from rear end of $\mathscr{A}_1(l+1)$ for producing $\mathscr{A}_1(l)$. For each deletion of $u_{t'}$ in $\mathscr{A}_1(l+1)$ we recompute the intersection of e with the perpendicular bisector of segment joining vertices v_l with a vertex attached with $u_{t'}$ (say u) only once. After the deletion step, the newly enumerated u appears at the left of the last element of the array and generates $\mathscr{A}_1(l)$. Observe that, at most $O(n)$ elements need to be deleted for computing $\{\mathscr{A}_1(l), l = n-1, n-2, \ldots, 1\}$. Thus the result follows. □

We now describe an iterative method for computing two minimum radii constrained circles $\mathscr{C}'$ and $\mathscr{C}''$ for covering all the vertices of P. Initially, we take $P_1(s) = \{v_0, v_{n-1}, v_{n-2}, \ldots, v_s\}$, where $x(v_i) \leqslant x(v_0)$ for all $i = n-1, n-2, \ldots, s$, and $P_2(s') = \{v_1, v_2, \ldots, v_{s'}\}$, where $x(v_i) \geqslant x(v_1)$ for all $i = 2, 3, \ldots, s'$. Let $\widehat{\mathscr{C}'}$ and $\widehat{\mathscr{C}''}$ are the minimum radii constrained circles for covering $P_1(s)$ and $P_2(s')$ respectively. Note that, the center of $\widehat{\mathscr{C}'}$ and $\widehat{\mathscr{C}''}$ are v_0 and v_1 respectively. Let the radii of $\widehat{\mathscr{C}'}$ and $\widehat{\mathscr{C}''}$ are r' and r'' respectively. At each iteration we do the following:

If $r' \geqslant r''$ and $\widehat{\mathscr{C}'}$ and $\widehat{\mathscr{C}''}$ do not cover all the vertices of P, then we execute the following steps to update $\widehat{\mathscr{C}''}$ such that it covers all the vertices of $P_2(s'+1) = \{v_1, v_2, \ldots, v_{s'}, v_{s'+1}\}$. Note that, the center of $\mathscr{C}''$ lies in the region $\mathscr{R}(v_{s'+1}) \cap e$. The center may be (i) the perpendicular projection of $v_{s'+1}$ on e, or (ii) a member of $\mathscr{A}_2(s'+1)$ which is computed as follows:

Prior to considering the vertex $v_{s'+1}$, let the circle $\widehat{\mathscr{C}''}$ passes through the vertex v_{s^*}, for some $s^* \leqslant s'$. If it passes through more than one vertex, then v_{s^*} is the rightmost one among them in the sequence $\{v_1, v_2, \ldots, v_{s'}\}$. We also have $\mathscr{A}_2(s^*) = \{u_0, u_1, \ldots, u_m\}$, where u_i is the intersection point of e and the perpendicular bisector of $[v_{s^i}, v_{s^{i-1}}]$ for some $s^{i-1} < s^i$. We compute the point of intersection (say u) of e and perpendicular bisector of $(v_{s'+1}, v_{s^{m-j}})$ for $j = 0, 1, \ldots$ until we have $x(u) > x(u_{m-j})$. Here, u is the desired center of circle $\widehat{\mathscr{C}''}$ as mentioned in Case (ii). We get $\mathscr{A}_2(s'+1)$ by removing all the members of $\mathscr{A}_2(s')$ whose x coordinate is less than $x(u)$, and finally add u in $\mathscr{A}_2(s'+1)$.

We set $r'' = d(u, v_{s'+1})$, where u is the center of $\widehat{\mathscr{C}''}$. Finally, if the updated $\widehat{\mathscr{C}''}$ covers some more vertices, namely $v_{s'+2}, v_{s'+3}, \ldots, v_{s'+j}$, then set $s^* = s'+1$, and $s' = s'+j$. In this case, we need not have to compute $\mathscr{A}_2(s'+2), \mathscr{A}_2(s'+3), \ldots, \mathscr{A}_2(s'+j)$.

If $r' < r''$ and $\widehat{\mathscr{C}'}$ and $\widehat{\mathscr{C}''}$ do not cover all the vertices of P, then we use $\mathscr{A}_1(s)$ to compute $\mathscr{A}_1(s-1)$, and follow the similar method as described above to update $\widehat{\mathscr{C}'}$ such that it covers all the vertices of $P_1(s-1) = \{v_0, v_{n-1}, v_{n-2}, \ldots, v_s, v_{s-1}\}$. After updating $\widehat{\mathscr{C}'}$, we compute its radius r'.

While considering the last uncovered vertex, a typical situation may arise. Let $\widehat{\mathscr{C}'}$ and $\widehat{\mathscr{C}''}$ be obtained after q iterative steps which cover all the vertices of P excepting only one vertex, say v, and their radii are r' and r'' respectively. Without loss of generality, assume that $r' \leqslant r''$. In the $q+1$th iteration, we include v in both $\widehat{\mathscr{C}'}$ and $\widehat{\mathscr{C}''}$. Let the corresponding radii are r^* and r^{**} respectively. If $r^* \geqslant r^{**}$, then $\mathscr{C}'$ is $\widehat{\mathscr{C}'}$ without including v, and $\mathscr{C}''$ is $\widehat{\mathscr{C}''}$ after including v. On the other hand, if $r^* < r^{**}$, then v is included in $\widehat{\mathscr{C}'}$. We also need to execute one more iteration as follows: let the $\widehat{\mathscr{C}''}$ is observed to pass through v_{s^*} in the q-th iteration and $r^* < r^{**}$. We include v_{s^*} in $\widehat{\mathscr{C}'}$, and observe its updated radius r^*. If $r^* < r''$, then $\mathscr{C}'$ is set to the updated $\widehat{\mathscr{C}'}$ in the $q+2$-th iteration (after including v_{s^*}). Otherwise, $\mathscr{C}'$ is set to the $\widehat{\mathscr{C}'}$ obtained in the $q+1$-th iteration. After fixing $\mathscr{C}'$, the remaining points will be covered by $\mathscr{C}''$.

Theorem 2. *Optimum size circles* $\mathscr{C}'$ *and* $\mathscr{C}''$ *with centers on edge e that cover all the vertices of polygon P can be located in* $O(n)$ *time.*

Proof. Result follows from above discussion and from Lemmata 7, 8 and 10. □

But there is no guarantee that the two circles $\mathscr{C}'$ and $\mathscr{C}''$ will cover the entire polygonal region.

Observation 6. *If* $\mathscr{C}'$ *and* $\mathscr{C}''$ *together do not cover the polygon P completely, then there exists exactly one edge* e', $(e' \neq e)$, *which is not fully covered by* $\mathscr{C}'$ *and* $\mathscr{C}''$ (*see* Fig. 7(a)).

If the situation stated in Observation 6 does not occur, then $\mathscr{C}_1 = \mathscr{C}'$ and $\mathscr{C}_2 = \mathscr{C}''$. Otherwise, the uncovered edge e' can be detected while computing $\mathscr{C}'$ and $\mathscr{C}''$. We now compute the optimum constrained circles $\mathscr{C}_1$ and $\mathscr{C}_2$.

Note that, $\mathscr{C}_1$ and $\mathscr{C}_2$ are of same size, and they must intersect at some point, say π, on edge e'. Let the equation of the line containing e' be $y = m \cdot x + c$. Let v_{s^*} be the vertex of P on the boundary of circle $\mathscr{C}'$, and having the least x-coordinate value, whereas $v_{s^{**}}$ be the vertex of P on the boundary of circle $\mathscr{C}''$, and having the maximum x-coordinate value among all such vertices. From Lemma 7, we can conclude that $\mathscr{C}_1$ passes through either vertex v_{s^*} or some other vertex to the left side of v_{s^*}. Similarly, $\mathscr{C}_2$ passes through either vertex $v_{s^{**}}$ or some other vertex to the right side of $v_{s^{**}}$. Our objective is to locate the point π and hence the centers α and β. The following observation guides us to detect these points.

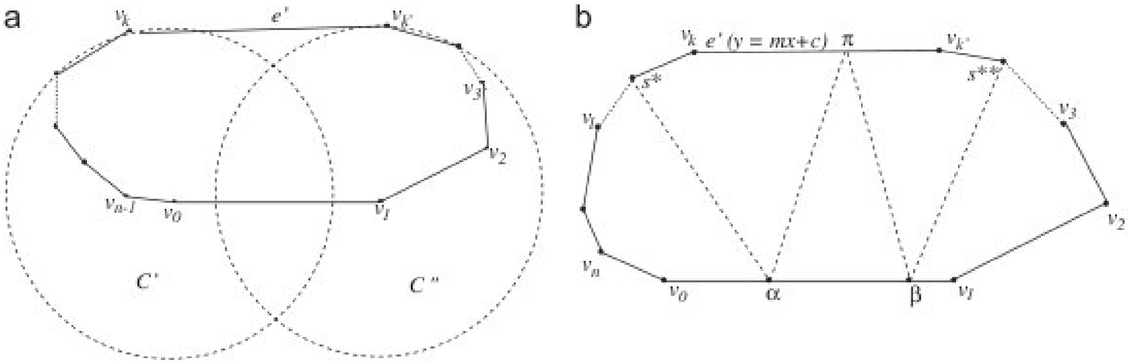

Fig. 7. Illustrating the proof of Observation 6.

Observation 7. *The point π is the intersection of the perpendicular bisector of the line segment $[\alpha, \beta]$ with the edge e'.*

Assume that, the coordinate of π is (x_π, y_π) and the coordinate of α is $(x_\alpha, 0)$. From Observation 7, we can say that the coordinate of β is $(2x_\pi - x_\alpha, 0)$ (see Fig. 7(b)). Initially, let us assume that $\mathscr{C}_1$ and $\mathscr{C}_2$ pass through v_{s^*} and $v_{s^{**}}$ respectively, whose coordinates are known. As both the circles pass through π, and have centers at α and β respectively, we can obtain a degree four polynomial equation involving x_π from the following constraints: (i) $d(v_{s^*}, \alpha) = d(\alpha, \pi)$, (ii) $d(v_{s^{**}}, \beta) = d(\beta, \pi)$, (iii) $y_\pi = m \cdot x_\pi + c$, and (iv) $x_\beta = 2x_\pi - x_\alpha$ (see Fig. 7(b)). Hence we can compute α, β and π in constant time. If $\alpha \notin \mathscr{R}(v_{s^*})$ then $\mathscr{C}_1$ passes through a vertex to the left of v_{s^*}. Without loss of generality, we choose v_{s^*+i}, and repeat the same procedure replacing v_{s^*} by v_{s^*+i} for $i = 1, 2, \ldots$ until $\alpha \in \mathscr{R}(v_{s^*+i})$. Similarly, if $\beta \notin \mathscr{R}(v_{s^{**}})$), we apply the same procedure assuming that $\mathscr{C}''$ passes through the vertex $v_{s^{**}-1}$.

Theorem 3. *The minimum radii constrained circles $\mathscr{C}_1$ and $\mathscr{C}_2$ covering the region P can be computed in $O(n)$ time.*

4. Conclusion

In real life, finding the location for placing mobile base station avoiding the forbidden region is an important problem in facility location. Suppose P be a polygonal region which is forbidden in order to place a base station in the context of mobile communication. Here, we consider the problem of placing one base station at any point on the boundary of P and assign a range such that every point in the region is covered by that base station and the range assigned to that base station for covering the region is minimum among all such possible choices of base stations. Here we consider the forbidden region P as convex and base station can be placed on the boundary of the region. We present optimum linear time algorithm for that problem. We also consider the placement problem for a pair of base stations on a specified side of the boundary such that the range assigned to those base stations in order to cover the region is minimum among all such possible choices of a pair of base stations on that side. We also present a linear time algorithm to solve this problem.

Acknowledgment

We are thankful to the anonymous reviewers whose comments helped to improve the presentation of our paper.

References

[1] A. Aggarwal, L.J. Guibas, J. Saxe, P. Shor, A linear time algorithm for computing the Voronoi diagram of a convex polygon, in: Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, pp. 39–45.
[3] A. Aggarwal, D. Kravets, A linear time algorithm for finding all furthest neighbours in a convex polygon, Inform. Process. Lett. (1989) 17–20.
[4] P.K. Agarwal, M. Sharir, E. Welzl, The discrete 2-center problem, in: Proceedings of the 13th Annual ACM Symposium on Computational Geometry, 1997, pp. 147–155.
[5] P. Bose, G. Toussaint, Computing the constrained Euclidean, geodesic and link center of a simple polygon with applications, in: Proceedings of the Pacific Graphics International, 1996, pp. 102–112.
[6] T.M. Chan, More planar two-center algorithms, Computational Geometry: Theory and Application 13 (1999) 189–198.
[7] J. Elzinga, D.W. Hearn, Geometrical solutions to some minimax location problems, Transportation Sci. 6 (1972) 379–394.
[8] F. Hurtado, V. Sacristan, G. Toussaint, Facility location problems with constraints, Stud. Locational Anal. 15 (2000) 17–35.
[9] S.K. Kim, C.-S. Shin, Efficient algorithms for two-center problems for a convex polygon, in: Proceedings of the 6th International Computing and Combinatorics Conference, Ban, NY, 2000, pp. 299–309.
[10] D.T. Lee, Furthest neighbour Voronoi diagrams and applications, Report 80-11-FC-04, Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, 1980.
[11] N. Megiddo, Linear-time algorithms for linear programming in R^3 and related problems, SIAM J. Comput. 12 (1983) 759–776.
[12] F. Preparata, Minimum spanning circle, in: Steps into Computational Geometry, Technical Report, University of Illinois, Urbana, IL, 1977.
[13] M.I. Shamos, D. Hoey, Closest-point problem, in: Proceedings of the 16th Annual IEEE Symposium on Foundations of Computer Science, 1975, pp. 151–162.
[14] M. Sharir, A near-linear algorithm for the planar 2-center problem, Discrete Comput. Geom. 18 (1997) 125–134.
[15] C. Shin, J. Kim, S.K. Kim, K. Chwa, Two-center problems for a convex polygon, in: Proceedings of the 6th Annual European Symposium on Algorithms, 1998, pp. 199–210.
[16] J.J. Sylvester, A question in the geometry of situation, Quart. J. Math. (1857) 1–79.

Sasanka Roy received his B.Sc in Mathematics (Hons.) from A.B.N. Seal College in 1998. He received his MCA from North Bengal University, Darjeeling in 2001. He has submitted his Ph.D. Thesis in Indian Statistical Institute under the supervision of Dr. Sandip Das. He is currently a retainer in Tata Research Development and Design Centre, Pune, India. His research interests are currently focused in the fields of Data Privacy, Computational Geometry, Algorithms, Facility location, Graph Theory.

Debabrata Bardhan received his Masters of Computer Application from the University of Burdwan of India in 2004. He is currently employed in LSI Technologies India Pvt. Ltd. His research interests are Computational Geometry, Storage system and Networks.

Sandip Das received his Bachelor of Statistics (Hons.), Master of Statistics, Master of Technology in Computer Science and Ph.D in Computer Science from the Indian Statistical Institute, Kolkata, in 1986, 1989, 1990 and 2001, respectively. From 1990 to 1999, he worked as a Lecturer in the Department of Computer Science, North Bengal University. Since 1999, he is working in the Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata. Currently he is an Associate Professor. His current research interests include graph theory, combinatorics, geometric algorithm, data structure etc.