Approximation algorithms for art gallery problems in polygons

Subir Kumar Ghosh
School of Computer Science, Tata Institute of Fundamental Research, Mumbai 400005, India

Abstract
In this paper, we present approximation algorithms for minimum vertex and edge guard problems for polygons with or without holes with a total of $n$ vertices. For simple polygons, approximation algorithms for both problems run in $O(n^4)$ time and yield solutions that can be at most $O(\log n)$ times the optimal solution. For polygons with holes, approximation algorithms for both problems give the same approximation ratio of $O(\log n)$, but the running time of the algorithms increases by a factor of $n$ to $O(n^5)$.

1. Introduction
The art gallery problem is to determine the number of guards that are sufficient to cover or see every point in the interior of an art gallery. An art gallery can be viewed as a polygon $P$ with or without holes with a total of $n$ vertices and guards as points in $P$. Any point $z \in P$ is said to be visible from a guard $g$ if the line segment joining $z$ and $g$ does not intersect the exterior of $P$. Usually the guards may be placed anywhere inside $P$. If the guards are restricted to vertices of $P$, we call them vertex guards. If there is no restriction, the guards are referred as point guards. Point and vertex guards are also referred as stationary guards. If the guards are mobile, i.e., able to patrol along a segment inside $P$, they are called mobile guards. If the mobile guards are restricted to edges of $P$, they are called edge guards.

The art gallery problem was first posed by Victor Klee for stationary guards (see [24]). Chvátal [9] proved that a simple polygon $P$ needs at most $\lfloor n/3 \rfloor$ stationary guards. Fisk [18] later gave a simple proof of this result using coloring technique, and based on his proof, Avis and Toussaint [3] developed an $O(n \log n)$ time algorithm for positioning guards in $P$. O’Rourke [35] showed that $P$ needs at most $\lfloor n/4 \rfloor$ mobile guards. For edge guards, $\lfloor n/4 \rfloor$ edge guards seem to be sufficient for guarding $P$, except for a few polygons (see [42]).

For a simple orthogonal polygon $P$, i.e., the edges of $P$ are horizontal or vertical, Kahn et al. [26] proved that $P$ needs at most $\lfloor n/4 \rfloor$ stationary guards. O’Rourke [34] later gave an alternative proof for this result. These proofs use the partition of $P$ into convex quadrilaterals before $\lfloor n/4 \rfloor$ guards are placed in $P$. Note that a convex quadrilaterization of $P$ can be obtained by algorithms of Edelsbrunner, O’Rourke and Welzl [12], Lubiw [32], Sack [38], and Sack and Toussaint [39]. Aggarwal [1] showed that $P$ needs at most $\lfloor \frac{3n+4}{16} \rfloor$ mobile guards. This bound also holds for edge guards as shown by Björling-Sachs [6].

For a polygon $P$ with $h$ holes, O’Rourke [36] showed that $P$ needs at most $\lfloor \frac{n+2h}{3} \rfloor$ vertex guards. Hoffmann, Kaufmann and Kriegel [23] and Björling-Sachs and Souvaine [7] proved independently that $P$ can always be guarded with $\lceil \frac{n+h}{3} \rceil$ point guards.

Note


E-mail address: ghosh@tifr.res.in.

1 The major part of the work was done when the author visited the Department of Electrical Engineering and Computer Science, The Johns Hopkins University during 1985–86 and was supported by NSF Grant DCR83-51468 and a grant from IBM.

0166-218X/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
guards. Bjorling-Sachs and Souvaine also gave an $O(n^3)$ time algorithm for positioning the guards. There is no tight bound known on the number of mobile guards required to guard $P$. Since $\left\lceil \frac{2n}{16h} \right\rceil$ point guards are sufficient to guard $P$, the bound naturally holds for mobile guards as well. To guard an orthogonal polygon $P$ with $h$ holes, Györi, Hoffmann, Kriegel and Shermer [22] proved that $\left\lceil \frac{2n}{16h} \right\rceil$ mobile guards are always sufficient to guard $P$. For the vertex guard problem, this restriction turns out to be a true restriction, as shown in the following lemma.

Lemma 2.1. Every convex component of $P$ is either totally visible or totally not visible from a vertex of $P$.
Let us assume on the contrary that there exists a convex component \( c \) which is partially visible from a vertex \( v_i \) of \( P \). As \( c \) is partially visible from \( v_i \), there exists a vertex \( v_j \) in \( P \) such that the line drawn from \( v_j \) through \( v_j \) intersects \( c \). And thus, \( c \) is not a convex component, a contradiction. \( \square \)

**Corollary 2.1.** For every vertex \( v_i \) of \( P \), the fan \( F_i \) is the union of some convex components of \( P \).

The above corollary suggests that the problem of finding the minimum number of fans to cover \( P \) is same as the minimum set-covering problem, where every fan is a set and convex components are elements of the set. The minimum set-covering problem can be stated as follows. Given a finite family of sets \( \{ S_1, \ldots, S_m \} \), the problem is to determine the minimum subset \( A \) of \( \{ S_1, \ldots, S_m \} \) such that \( \bigcup S_i \in A = \bigcup_{i=1}^{n} S_i \). For details on the minimum set-covering problem, see Garey and Johnson [19], and Vazirani [43]. There are many approximation algorithms for the minimum set-covering problem [4,10,25,31], and here we use Johnson’s approximation algorithm [25]. In the following, we present the algorithm for locating vertex guards in \( P \).

**Vertex guard algorithm:**

1. **Step 1** Draw lines through every pair of vertices of \( P \) and compute all convex components \( c_1, c_2, \ldots, c_m \) of \( P \). Let \( C = (c_1, c_2, \ldots, c_m) \), \( N = (1, 2, \ldots, n) \) and \( Q = \emptyset \).

2. **Step 2** For \( 1 \leq j \leq n \), construct the set \( F_j \) by adding those convex components of \( P \) that are totally visible from the vertex \( v_j \).

3. **Step 3** Find \( i \in N \) such that \( |F_i| \geq |F_j| \) for all \( j \in N \) and \( i \neq j \).

4. **Step 4** Add \( i \) to \( Q \) and delete \( i \) from \( N \).

5. **Step 5** For all \( j \in N \), \( F'_j := F_j - F_i \), and \( C := C - F_i \).

6. **Step 6** If \( |C| \neq 0 \) then goto **Step 3**.

7. **Step 7** Output the set \( Q \) and Stop.

Let us analyze the time complexity of the algorithm. Since \( O(n^2) \) lines are drawn in \( P \) to compute convex components, \( m \) can be at most \( O(n^4) \), and thus **Step 1** requires \( O(n^4) \) time. Constructing \( F_1, F_2, \ldots, F_n \) in **Step 2** can be done as follows. For every convex component \( c_k \) of \( C \), take a point \( z_k \in c_k \) and compute \( VP(P, z_k) \). For every vertex \( v_j \in VP(P, z_k) \), add \( c_j \) to \( F_j \). Repeat this process for all \( k \). If \( P \) is a simple polygon, \( VP(P, z_k) \) can be computed in \( O(n) \) time by the algorithm of Lee [29]. If \( P \) is a polygon with holes, \( VP(P, z_k) \) can be computed in \( O(n) \) time by the query algorithm of Asano et al. [2] after spending \( O(n^2) \) time for preprocessing. Therefore, **Step 2** takes \( O(nm) \) time. **Step 3** requires \( O(n^2) \) time. For every convex component \( c_i \in F_i \), we know the fans containing \( c_i \). Consequently, removing \( c_i \) from all sets containing \( c_i \) in **Step 5** can be done in \( O(n) \) time. Since every convex component is considered once, the entire process of removing all convex components in **Step 5** takes \( O(nm) \) time. Hence the overall time complexity of the approximation algorithm is \( O(n^2) \).

The above analysis holds for polygons \( P \) with holes. If \( P \) is a simple polygon, the running time of the algorithm reduces to \( O(n^2) \) as the number of convex components \( m \) is \( O(n^3) \) as follows. In **Step 1** of the algorithm, convex components of \( P \) are computed by drawing lines passing through every pair of vertices of \( P \). Instead, for every vertex \( v_i \), \( VP(P, v_i) \) is computed by the algorithm of Lee [29]. We know that the boundary of \( VP(P, v_i) \) consists of polygonal edges and constructed edges [20]. Suppose all constructed edges of every \( VP(P, v_i) \) are added to \( P \). It can be seen that \( P \) is decomposed into convex components by these constructed edges. For details, see Bose, Lubiw and Munro [8]. Observe that since \( P \) is a simple polygon, a constructed edge of \( VP(P, v_i) \) can be intersected by only two constructed edges of \( VP(P, v_j) \) for all \( j \) except \( i \). Therefore, the total number intersection points on any constructed edge of \( VP(P, v_i) \) can be at most \( O(n) \). So, there can be at most \( O(n^2) \) convex components in \( P \). As in both cases the size of the input is \( O(nm) \) Johnson [25] has shown that \( |Q| \leq c_{opt} \cdot O(\log n) \).

We have the following theorems.

**Theorem 2.1.** For a simple polygon \( P \) of \( n \) vertices, an approximate solution of the minimum vertex guard problem can be computed in \( O(n^4) \) time and the size of the solution is at most \( O(\log n) \) times the optimal.
Theorem 2.2. For a polygon $P$ with holes with a total of $n$ vertices, an approximate solution of the minimum vertex guard problem can be computed in $O(n^2)$ time and the size of the solution is at most $O(\log n)$ times the optimal.

3. Approximation algorithms for edge guards

Assume that edges of the given polygon $P$ are labeled $e_1, e_2, \ldots, e_n$. A point $z \in P$ is said to be weakly visible from an edge $e_i$ of $P$ if there exists a point $u$ on $e_i$ such that the segment $zu$ lies inside $P$. Let $VP(P, e_i)$ denote the set of all points of $P$ that are weakly visible from $e_i$. We call $VP(P, e_i)$ the weak visibility polygon of $P$ from $e_i$. Since the region of $P$ that can be seen by an edge guard is a weak visibility polygon of $P$, the edge guard problem of $P$ can be viewed as a polygon decomposition problem in which decomposed pieces are weak visibility polygons. As before, $P$ is decomposed into convex components by drawing lines passing through every pair of vertices of $P$. We have the following lemma.

Lemma 3.1. Every convex component of $P$ is either totally visible or totally not visible from an edge of $P$.

Proof. Assume on the contrary that there exists a convex component $c$ which is partially visible from an edge $e_i$ of $P$. As $c$ is partially visible from $e_i$, there exist two points $z_1 \in c$ and $z_2 \in c$ such that one point (say, $z_1$) is visible from some point on $e_i$, but the other point $z_2$ is not visible from any point of $e_i$. Consequently, there exists a constructed edge of $VP(P, e_i)$ that has intersected the segment $z_1z_2$. Since every constructed edge of $VP(P, e_i)$ is a part of the segment drawn through two vertices of $P$, both $z_1$ and $z_2$ cannot belong to the same convex component $c$, a contradiction.

Corollary 3.1. For every edge $e_i$ of $P$, the weak visibility polygon of $P$ from $e_i$ is the union of some convex components of $P$.

The above corollary suggests that every $VP(P, e_i)$ can be viewed as a set (denoted as $E_i$) and convex components are elements of the sets. We present below the algorithm for locating edge guards in $P$ following the vertex guard algorithm stated in the previous section.

Edge guard algorithm:

Step 1 Draw lines through every pair of vertices of $P$ and compute all convex components $c_1, c_2, \ldots, c_m$ of $P$. Let $C = (c_1, c_2, \ldots, c_m)$, $N = (1, 2, \ldots, n)$ and $Q = \emptyset$.

Step 2 For $1 \leq j \leq n$, construct the set $E_j$ by adding those convex components of $P$ that are totally visible from the edge $e_j$ of $P$.

Step 3 Find $i \in N$ such that $|E_i| \geq |E_j|$ for all $j \in N$ and $i \neq j$.

Step 4 Add $i$ to $Q$ and delete $i$ from $N$.

Step 5 For all $j \in N$, $E_j := E_j - E_i$, and $C := C - E_i$.

Step 6 If $|C| \neq 0$ then goto Step 3.

Step 7 Output the set $Q$ and Stop.

The time complexity of the algorithm and the upper bound are stated in the following theorems. Note that for constructing sets $E_1, E_2, \ldots, E_n$ in Step 2, the same method of computing $VP(P, z_k)$ is used as stated in the previous section. Only difference is that $c_k$ is added to $E_j$ if any point of $e_j$ belongs to $VP(P, z_k)$. Also note that the sizes of $C$ in this algorithm for polygons with holes and polygons without holes are different by a factor of $n$.

Theorem 3.1. For a polygon $P$ with holes with a total of $n$ vertices, an approximate solution of the minimum edge guard problem can be computed in $O(n^2)$ time and the size of the solution is at most $O(\log n)$ times the optimal.

Theorem 3.2. For a simple polygon $P$ of $n$ vertices, an approximate solution of the minimum edge guard problem can be computed in $O(n^4)$ time and the size of the solution is at most $O(\log n)$ times the optimal.

4. Concluding remarks

The worst-case behavior of Johnson’s approximation algorithm [25] for the minimum set-covering problem is not very attractive. Lovasz [31] proved that any greedy algorithm for this problem cannot guarantee a better bound. In our case, the upper bound of $O(\log n)$ may not be reached because of certain geometric constraints. In the minimum set-covering problem, any subset of elements can form a set, which is not true in our case as elements of a set (i.e. convex components) have to form a polygonal region corresponding to a visibility polygon of $P$ from a vertex or an edge. This geometric restriction on our input sets may not allow the approximation ratio of our approximation algorithms to reach the upper bound of $O(\log n)$. It is open if a tighter upper bound on the approximation ratio can be obtained for our approximation algorithms by taking polygonal geometry into consideration.

Regarding the lower bound on the approximation ratio for the problems of minimum vertex, point and edge guards in simple polygons, Eidenbenz, Stamm and Widmayer [15] showed that these problems are APX-hard. This means that for each of these problems, there exists a constant $\epsilon > 0$ such that an approximation ratio of $1 + \epsilon$ cannot be guaranteed by any polynomial time approximation algorithm unless $P = NP$. Though there may be approximation algorithms for
these problems whose approximation ratios are not small constants, for polygons with holes, these problems cannot be approximated by a polynomial time algorithm with ratio \((1 - \epsilon)/12\)\((\ln n)\) for any \(\epsilon > 0\), unless \(NP \subseteq TIME(n^{O(\log \log n)})\). The results are obtained by using gap-preserving reductions from the SET COVER problem. Therefore, the open problem is to design approximation algorithms for vertex, edge and point guards problems in simple polygons which yield solutions within a constant factor of the optimal solution.

Acknowledgements

The author gratefully acknowledges helpful comments and suggestions of Joseph O’Rourke during the preparation of a preliminary version of this paper which was circulated as a report (JHU/EECS-86/15, August 1986) of the Department of Electrical Engineering and Computer Science, The Johns Hopkins University. The author would like to thank Sudeb Pal for his helpful comments during the preparation of this version of the paper.

In recent years, the author received several requests from young researchers for sending a copy of a preliminary version of the paper as the Proceedings of the Canadian Information Processing Society Congress, 1987 is not easily accessible. These requests have motivated the author for preparing the journal version of the paper after two decades. The author would like to thank these researchers for their interest in the paper.

References