The Quest for Optimal Solutions for the Art Gallery Problem: A Practical Iterative Algorithm

Pritam Bhattacharya

Department of Computer Science and Engineering,
Indian Institute of Technology, Kharagpur

(Based on work by Tozoni, de Rezende, de Souza)

Polygons and Visibility

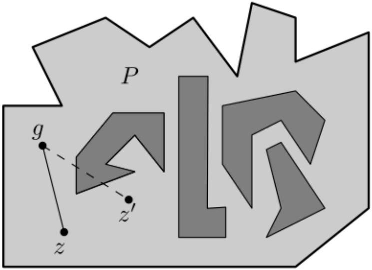


Figure: Polygon with holes

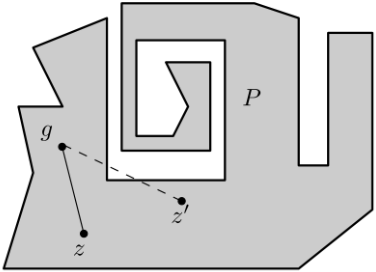


Figure: Polygon without holes

Definition (Visibility of a Point)

Any point $z \in P$ is said to be *visible* from another point $g \in P$ if the line segment zg does not intersect the exterior of P.

Art Gallery Problem

The art gallery problem (AGP) enquires about the least number of guards that are sufficient to ensure that an art gallery is fully guarded, assuming that a guard's field of view covers 360° as well as an unbounded distance.

Art Gallery Problem

The art gallery problem (AGP) enquires about the least number of guards that are sufficient to ensure that an art gallery is fully guarded, assuming that a guard's field of view covers 360° as well as an unbounded distance.

An art gallery can be viewed as an n-sided polygon P (with or without holes) and guards as points in P.

Art Gallery Problem

The art gallery problem (AGP) enquires about the least number of guards that are sufficient to ensure that an art gallery is fully guarded, assuming that a guard's field of view covers 360° as well as an unbounded distance.

An art gallery can be viewed as an n-sided polygon P (with or without holes) and guards as points in P.

Guards may be allowed to be placed anywhere within P (*point guards*), or they may be allowed to be placed only on the vertices of P (*vertex guards*).

Art Gallery Problem

The art gallery problem (AGP) enquires about the least number of guards that are sufficient to ensure that an art gallery is fully guarded, assuming that a guard's field of view covers 360° as well as an unbounded distance.

An art gallery can be viewed as an n-sided polygon P (with or without holes) and guards as points in P.

Guards may be allowed to be placed anywhere within P (*point guards*), or they may be allowed to be placed only on the vertices of P (*vertex guards*).

Victor Klee (1973) $\rightarrow$ How many point guards or vertex guards are always sufficient to guard a simple polygon having n vertices?

Sufficient Number of Guards

Theorem (Chvatal (1975), Fisk (1978))

For guarding a simple polygon with n vertices, $\lfloor \frac{n}{3} \rfloor$ point guards or vertex guards are sufficient and sometimes necessary.

Sufficient Number of Guards

Theorem (Chvatal (1975), Fisk (1978))

For guarding a simple polygon with n vertices, $\lfloor \frac{n}{3} \rfloor$ point guards or vertex guards are sufficient and sometimes necessary.

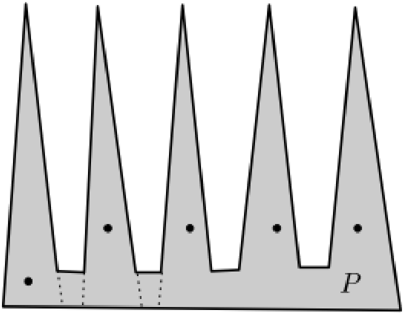


Figure: A polygon where $\lfloor \frac{n}{3} \rfloor$ point guards or vertex guards are necessary.

Art Gallery Problem - Hardness Results

Definition (Decision Version of the Art Gallery Problem)

Given a polygon P and a number k as input, can the polygon P be guarded with k or fewer guards?

Art Gallery Problem - Hardness Results

Definition (Decision Version of the Art Gallery Problem)

Given a polygon P and a number k as input, can the polygon P be guarded with k or fewer guards?

Hardness results known for the decision version of AGP:

- Proved to be NP-complete for vertex guards (Lee and Lin).

Art Gallery Problem - Hardness Results

Definition (Decision Version of the Art Gallery Problem)

Given a polygon P and a number k as input, can the polygon P be guarded with k or fewer guards?

Hardness results known for the decision version of AGP:

- Proved to be NP-complete for vertex guards (Lee and Lin).
- Proved to be NP-complete for point guards (Aggarwal).

Art Gallery Problem - Hardness Results

Definition (Decision Version of the Art Gallery Problem)

Given a polygon P and a number k as input, can the polygon P be guarded with k or fewer guards?

Hardness results known for the decision version of AGP:

- Proved to be NP-complete for vertex guards (Lee and Lin).
- Proved to be NP-complete for point guards (Aggarwal).
- Proved to be APX-complete (Eidenbenz, Stamm and Widmayer), implying that no PTAS can exist for AGP.

Art Gallery Problem - Hardness Results

Definition (Decision Version of the Art Gallery Problem)

Given a polygon P and a number k as input, can the polygon P be guarded with k or fewer guards?

Hardness results known for the decision version of AGP:

- Proved to be NP-complete for vertex guards (Lee and Lin).
- Proved to be NP-complete for point guards (Aggarwal).
- Proved to be APX-complete (Eidenbenz, Stamm and Widmayer), implying that no PTAS can exist for AGP.
- Specifically for polygons with holes, AGP cannot be approximated to within a factor of $\Omega(\ln n)$ (Eidenbenz, Stamm and Widmayer).

Art Gallery Problem - Approximation Algorithms

For computing the minimum number of guards, the following approximation algorithms exist:

- $\mathcal{O}(\log n)$-approximation algorithm for vertex and edge guards by Ghosh in 1987 via a reduction to set cover.

Art Gallery Problem - Approximation Algorithms

For computing the minimum number of guards, the following approximation algorithms exist:

- $\mathcal{O}(\log n)$-approximation algorithm for vertex and edge guards by Ghosh in 1987 via a reduction to set cover.
- $\mathcal{O}(\log OPT)$-approximation pseudopolynomial time algorithm for point guards and perimeter guards by Deshpande et al.

Art Gallery Problem - Approximation Algorithms

For computing the minimum number of guards, the following approximation algorithms exist:

- $\mathcal{O}(\log n)$-approximation algorithm for vertex and edge guards by Ghosh in 1987 via a reduction to set cover.
- $\mathcal{O}(\log OPT)$-approximation pseudopolynomial time algorithm for point guards and perimeter guards by Deshpande et al.
- $\mathcal{O}(\log OPT)$-approximation randomized algorithm with fully polynomial expected running time by Efrat and Har-Peled.

Art Gallery Problem - Approximation Algorithms

For computing the minimum number of guards, the following approximation algorithms exist:

- $\mathcal{O}(\log n)$-approximation algorithm for vertex and edge guards by Ghosh in 1987 via a reduction to set cover.
- $\mathcal{O}(\log OPT)$-approximation pseudopolynomial time algorithm for point guards and perimeter guards by Deshpande et al.
- $\mathcal{O}(\log OPT)$-approximation randomized algorithm with fully polynomial expected running time by Efrat and Har-Peled.
- $\mathcal{O}(\log \log OPT)$-approximation algorithm for perimeter guards by King and Kirkpatrick in 2011 by using ϵ-nets.

Art Gallery Problem - Approximation Algorithms

For computing the minimum number of guards, the following approximation algorithms exist:

- $\mathcal{O}(\log n)$-approximation algorithm for vertex and edge guards by Ghosh in 1987 via a reduction to set cover.
- $\mathcal{O}(\log OPT)$-approximation pseudopolynomial time algorithm for point guards and perimeter guards by Deshpande et al.
- $\mathcal{O}(\log OPT)$-approximation randomized algorithm with fully polynomial expected running time by Efrat and Har-Peled.
- $\mathcal{O}(\log \log OPT)$-approximation algorithm for perimeter guards by King and Kirkpatrick in 2011 by using ϵ-nets.

Conjecture (Ghosh (1987))

There exist polynomial time algorithms with a constant approximation ratio for vertex guarding polygons without holes.

Importance of this Work

- Presents a practical iterative algorithm for the Art Gallery Problem with point guards, which finds a sequence of decreasing upper bounds and increasing lower bounds for the optimal value.

Importance of this Work

- Presents a practical iterative algorithm for the Art Gallery Problem with point guards, which finds a sequence of decreasing upper bounds and increasing lower bounds for the optimal value.
- As evidence of effectiveness of the proposed algorithm, presents results showing that for every one of more than 1440 benchmark polygons of various classes gathered from the literature with up to a thousand vertices, optimal solutions are attained in just a few minutes of computing time.

Importance of this Work

- Presents a practical iterative algorithm for the Art Gallery Problem with point guards, which finds a sequence of decreasing upper bounds and increasing lower bounds for the optimal value.
- As evidence of effectiveness of the proposed algorithm, presents results showing that for every one of more than 1440 benchmark polygons of various classes gathered from the literature with up to a thousand vertices, optimal solutions are attained in just a few minutes of computing time.
- This work is unprecedented since, despite several decades of extensive investigation on the AGP, all previously published algorithms were unable to handle instances of that size and often failed to prove optimality for a significant fraction of the instances tested.

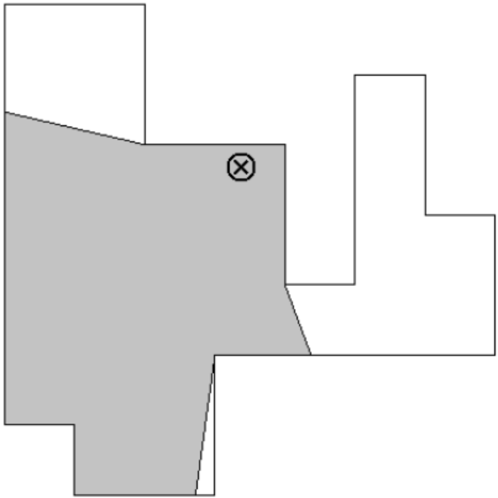
Visibility Polygon

Definition (Visibility Polygon)

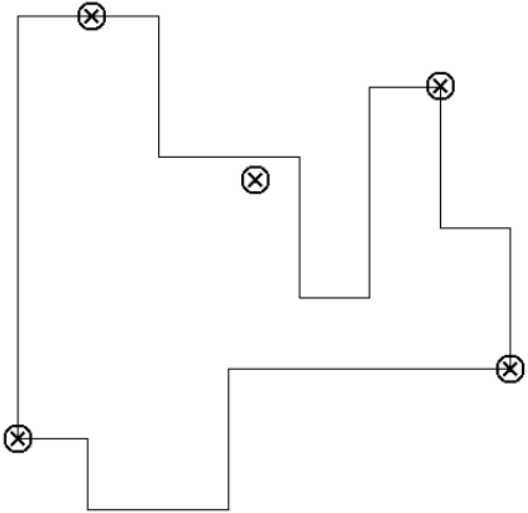
The visibility polygon of a point $p \in P$, denoted by $Vis(p)$, is the set of all points in P that are visible from p.

The edges of $Vis(p)$ are called *visibility edges*, and they are said to be *proper* for p if and only they are not contained in any edge of P.

Visibility Polygon



Atomic Visibility Polygons



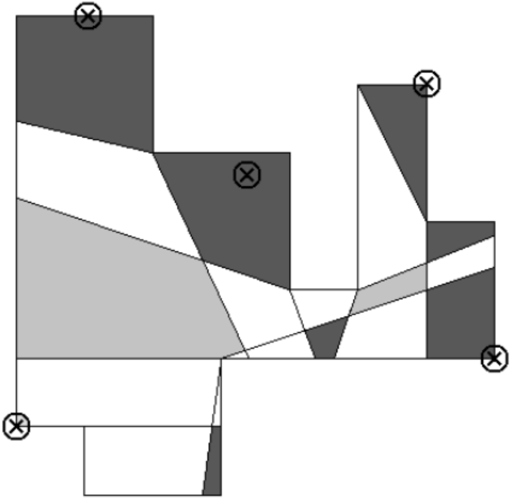
Atomic Visibility Polygons

Definition (Atomic Visibility Polygons)

The geometric arrangement defined by the visibility edges of the points in S partitions P into a collection of convex polygonal faces called Atomic Visibility Polygons or simply AVPs.

Clearly, the edges of an AVP are either portions of edges of P or portions of proper visibility edges for points of S.

Atomic Visibility Polygons



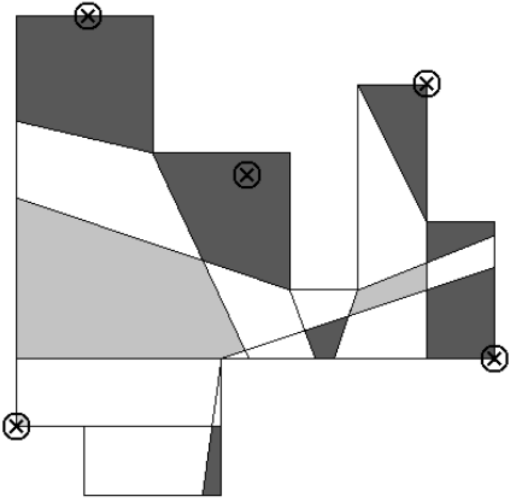
Atomic Visibility Polygons - Light and Shadow

AVPs can be classified according to their visibility properties relative to the points of S.

Definition (Light and Shadow AVPs)

We say that an AVP $\mathcal{F}$ is a light (shadow) AVP if there exists a subset T of S such that $\mathcal{F}$ is (is not) visible from any point in T and the only proper visibility edges that spawn $\mathcal{F}$ emanate from points in T.

Atomic Visibility Polygons



Discretized Versions of AGP - AGPFC

In the *Art Gallery Problem With Fixed Guard Candidates* (AGPFC), one is given a finite set of points $C \subset P$, and the question consists of selecting the minimum number of guards in C that are sufficient to cover the entire polygon.

Discretized Versions of AGP - AGPFC

In the *Art Gallery Problem With Fixed Guard Candidates* (AGPFC), one is given a finite set of points $C \subset P$, and the question consists of selecting the minimum number of guards in C that are sufficient to cover the entire polygon.
A special case of the AGPFC is obtained when the elements of C are restricted to the vertices of P, in which case we call it the *Art Gallery Problem With Vertex Guards* (AGPVG).

Discretized Versions of AGP - AGPW

In the *Art Gallery Problem With Witnesses* (AGPW), one is given a finite set of points $W \subset P$, and the problem consists in finding the minimum number of guards in P that are sufficient to cover all points in W.

Discretized Versions of AGP - AGPW

In the *Art Gallery Problem With Witnesses* (AGPW), one is given a finite set of points $W \subset P$, and the problem consists in finding the minimum number of guards in P that are sufficient to cover all points in W.

Clearly, coverage of W does not ensure that of P. A polygon P to be *witnessable* if there exists a finite witness set $W \subset P$ satisfying the property that any set of guards that covers W also covers the entire polygon P.

Discretized Versions of AGP - AGPWFC

If both the witness set and the guard candidate set are required to be finite, then the corresponding discretization leads to a hybrid of the last two problems, which we will denote by AGPWFC.

Discretized Versions of AGP - AGPWFC

If both the witness set and the guard candidate set are required to be finite, then the corresponding discretization leads to a hybrid of the last two problems, which we will denote by AGPWFC.
It is worth noting that the latter problem can easily be cast as a Set Cover Problem (SCP) in which the elements of W have to be covered using the subsets comprised of the witness points that are covered by the candidate guards.

Discretized Versions of AGP - AGPWFC

If both the witness set and the guard candidate set are required to be finite, then the corresponding discretization leads to a hybrid of the last two problems, which we will denote by AGPWFC.
It is worth noting that the latter problem can easily be cast as a Set Cover Problem (SCP) in which the elements of W have to be covered using the subsets comprised of the witness points that are covered by the candidate guards.
Despite being NP-hard, large instances of the SCP can be solved quite efficiently using modern integer programming solvers.

Notations used in the Algorithm

- Let V denote the set of vertices of the input polygon P and assume that $|V| = n$.

Notations used in the Algorithm

- Let V denote the set of vertices of the input polygon P and assume that $|V| = n$.
- Given a finite set S of points in P, we denote by $Arr(S)$ the arrangement defined by the visibility edges of the points in S.

Notations used in the Algorithm

- Let V denote the set of vertices of the input polygon P and assume that $|V| = n$.
- Given a finite set S of points in P, we denote by $Arr(S)$ the arrangement defined by the visibility edges of the points in S.
- Let $C_{\mathcal{U}}(S)$ be a set comprised of one point from the interior of each uncovered region induced by S in P.

Notations used in the Algorithm

- Let V denote the set of vertices of the input polygon P and assume that $|V| = n$.
- Given a finite set S of points in P, we denote by $Arr(S)$ the arrangement defined by the visibility edges of the points in S.
- Let $C_{\mathcal{U}}(S)$ be a set comprised of one point from the interior of each uncovered region induced by S in P.
- We denote by $V_{\mathcal{L}}(S)$ the set of vertices of the light AVPs of $Arr(S)$.

Notations used in the Algorithm

- Let V denote the set of vertices of the input polygon P and assume that $|V| = n$.
- Given a finite set S of points in P, we denote by $Arr(S)$ the arrangement defined by the visibility edges of the points in S.
- Let $C_{\mathcal{U}}(S)$ be a set comprised of one point from the interior of each uncovered region induced by S in P.
- We denote by $V_{\mathcal{L}}(S)$ the set of vertices of the light AVPs of $Arr(S)$.
- We denote by $C_S(S)$ the set of centroids of the shadow AVPs of $Arr(S)$.

Notations used in the Algorithm

- Let D and C denote, respectively, a finite witness set and a finite candidate guard set.

Notations used in the Algorithm

- Let D and C denote, respectively, a finite witness set and a finite candidate guard set.
- Let $AGPW(D)$ indicate the AGP with witness set D.

Notations used in the Algorithm

- Let D and C denote, respectively, a finite witness set and a finite candidate guard set.
- Let $AGPW(D)$ indicate the AGP with witness set D.
- Let $AGPFC(C)$ indicate the AGP with candidate guard set C.

Notations used in the Algorithm

- Let D and C denote, respectively, a finite witness set and a finite candidate guard set.
- Let $AGPW(D)$ indicate the AGP with witness set D.
- Let $AGPFC(C)$ indicate the AGP with candidate guard set C.
- Lastly, $AGPWFC(D, C)$ refers to the AGP with witness set D and candidate guard set C.

Computing Lower Bounds

Theorem

Let D be a finite subset of points in P. Then, there exists an optimal solution for $AGPW(D)$ where each guard belongs to a light AVP of $Arr(D)$.

Computing Lower Bounds

Theorem

Let D be a finite subset of points in P. Then, there exists an optimal solution for $AGPW(D)$ where each guard belongs to a light AVP of $Arr(D)$.

Corollary

An optimal solution for $AGPW(D)$ can be obtained simply by solving $AGPWFC(D, V_{\mathcal{L}}(D))$.

Computing Lower Bounds

Theorem

Let D be a finite subset of points in P. Then, there exists an optimal solution for $AGPW(D)$ where each guard belongs to a light AVP of $Arr(D)$.

Corollary

An optimal solution for $AGPW(D)$ can be obtained simply by solving $AGPWFC(D, V_{\mathcal{L}}(D))$.

Corollary

Since D is a subset of points of P, the optimum of $AGPW(D)$ is a lower bound for the optimum value of the AGP on P.

Computing Upper Bounds

Theorem

Let D and C be two finite subsets of P, such that C covers P. Assume that $G(D, C)$ is an optimal solution for $AGPWFC(D, C)$. If $G(D, C)$ covers P, then $G(D, C)$ is also an optimal solution for $AGPFC(C)$.

Computing Upper Bounds

Theorem

Let D and C be two finite subsets of P, such that C covers P. Assume that $G(D, C)$ is an optimal solution for $AGPWFC(D, C)$. If $G(D, C)$ covers P, then $G(D, C)$ is also an optimal solution for $AGPFC(C)$.

Corollary

Since C is a subset of points of P, $|G(D, C)|$ is an upper bound for the optimum value of the AGP on P.

Computing Upper Bounds

Theorem

Let D and C be two finite subsets of P, such that C covers P. Assume that $G(D, C)$ is an optimal solution for $AGPWFC(D, C)$. If $G(D, C)$ covers P, then $G(D, C)$ is also an optimal solution for $AGPFC(C)$.

Corollary

Since C is a subset of points of P, $|G(D, C)|$ is an upper bound for the optimum value of the AGP on P.

Suppose $|G(D, C)|$ is not a valid upper bound for the AGP.
Then, the witness set D is updated to $D \cup C_{\mathcal{U}}(G(D, C))$.
This process is repeated until $G(D, C)$ covers P.

Pseudocode for the Algorithm

Algorithm 1 AGP Algorithm

$D \leftarrow$ initial witness set {see paragraph 4.2}
Set: $LB \leftarrow 0$, $UB \leftarrow n$ and $G^* \leftarrow V$
loop
    Solve AGPW(D) : set $G_w \leftarrow$ optimal solution and $z_w \leftarrow |G_w|$
    $C \leftarrow V_{\mathcal{C}}(D) \cup V$
    if G_w is a coverage of P **then**
        return G_w
    else
        $U \leftarrow C_{\mathcal{U}}(G_w)$
        $LB \leftarrow \max\{LB, z_w\}$ {Theorem 1}
    end if
    if $LB = UB$ **then**
        return G^*
    end if
    $D_f \leftarrow D \cup U$
    repeat
        Solve AGPWFC(D_f, C) : set $G_f \leftarrow$ optimal solution and $z_f \leftarrow |G_f|$
        if G_f is a coverage of P **then**
            $UB \leftarrow \min\{UB, z_f\}$ and, if $UB = z_f$, set $G^* \leftarrow G_f$ {Theorem 2}
        else
            $D_f \leftarrow D_f \cup C_{\mathcal{U}}(G_f)$
        end if
    until G_f is a coverage of P
    if $LB = UB$ **then**
        return G_f
    else
        $D \leftarrow D \cup U \cup M$ {M: see paragraph 4.3}
    end if
end loop

Initial Witness Set

- All-Vertices (AV)

Initial Witness Set

- All-Vertices (AV)
- Convex-Vertices (CV)

Initial Witness Set

- All-Vertices (AV)
- Convex-Vertices (CV)
- Chwa-Points (CP)
  (midpoints of all reflex-reflex edges and all convex vertices from convex-reflex edges)

Initial Witness Set

- All-Vertices (AV)
- Convex-Vertices (CV)
- Chwa-Points (CP)
  (midpoints of all reflex-reflex edges and all convex vertices from convex-reflex edges)
- Chwa-Extended (CE)
  (the same points as in CP plus all reflex vertices from convex-reflex edges)

Experimental Results

- All tests were conducted using a single desktop PC featuring an Intel CoreTM i7-2600 at 3.40 GHz, 8 GB of RAM and running under GNU/Linux 3.2.0.
- CGAL and XPRESS libraries were used in the C++ implementation.
- All tests were run in isolation, meaning that no other processes were executed at the same time on the machine.

Experimental Results

Instance Groups	n	Optimality Rates	
		Kröller et al.	Our Method
Simple (30 inst. per size)	60	80%	100%
	100	64%	100%
	200	44%	100%
	500	4%	100%
Orthogonal (30 inst. per size)	60	80%	100%
	100	54%	100%
	200	19%	100%
	500	7%	100%

Instance Groups	n	Time (sec)	
		Kröller et al.	Our Method
Simple (30 inst. per size)	60	0.70	0.57
	100	29.40	1.72
	200	14.90	7.09
	500	223.30	65.64
Orthogonal (30 inst. per size)	60	0.40	0.30
	100	1.10	0.95
	200	4.30	3.95
	500	25.30	30.85

Figure: Comparison with the method of Kroller et al.

Experimental Results

Instance Groups	n	Number of Guards (average) Bottino et al.	Number of Guards (average) Our Method
Simple (20 inst. per size)	30	4.20	4.20
	40	5.60	5.55
	50	6.70	6.60
	60	8.60	8.35
Orthogonal (20 inst. per size)	30	4.60	4.52
	40	6.10	6.00
	50	7.80	7.70
	60	9.30	9.10

Instance Groups	n	Time (sec) Bottino et al.	Time (sec) Our Method
Simple (20 inst. per size)	30	1.57	0.17
	40	2.97	0.23
	50	221.92	0.42
	60	271.50	0.54
Orthogonal (20 inst. per size)	30	1.08	0.12
	40	9.30	0.17
	50	6.41	0.23
	60	81.95	0.30

Figure: Comparison with the method of Bottino et al.

Thank You!